Requested Patent: JP2000215168A

Title:

AUTHENTICATION AND ACCESS CONTROL IN A MANAGEMENT CONSOLE PROGRAM FOR MANAGING SERVICES IN A COMPUTER NETWORK ;

Abstracted Patent: EP0977399 ;

Publication Date: 2000-02-02 ;

Inventor(s):

CHANG APRIL S (US); SNYDER ALAN (US); LARGE ANDREW R (US) ;

Applicant(s): SUN MICROSYSTEMS INC (US) ;

Application Number: EP19990305924 19990726 ;

Priority Number(s): US19980124181 19980728 ;

IPC Classification: H04L12/24 ; H04L29/06 ;

Equivalents: US6157953

ABSTRACT:

A method and apparatus of securing access to a service manager for the administration of services residing on multiple service host computers from an administration server computer is described. A user identifier, such as a user name, and a corresponding password are provided to the service manager. The user identifier is associated with a system administrator having administrative access to the services. The service manager authenticates the user by comparing the user identifier and password against a list of user identifiers and corresponding passwords stored in a persistent memory. A list of services to which the system administrator has administrative access is derived from the data in persistent memory. When the system administrator makes a request to administer one or more services from the list of services, the administrator's access control is verified at the service host computers on which the requested services reside by examining access control data in the persistent memory. Management files are transferred from the service host computers to the administration server computer thereby facilitating manipulation of the management files utilizing the service manager.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2000－215168
（P2000－215168A）
(43)公開日　平成12年８月４日(2000.8.4)

(51)Int.Cl.'	識別記号	ＦＩ		テーマコード'(参考)
Ｇ０６Ｆ　15/00	３３０	Ｇ０６Ｆ　15/00	３３０Ｂ	
Ｇ０９Ｃ　1/00	６６０	Ｇ０９Ｃ　1/00	６６０Ｅ	
Ｈ０４Ｌ　9/32		Ｈ０４Ｌ　9/00	６７３Ａ	
			６７５Ａ	

審査請求　未請求　請求項の数15　ＯＬ　外国語出願　（全 75 頁）

(21)出願番号　特願平11－214030

(22)出願日　平成11年７月28日(1999.7.28)

(31)優先権主張番号　０９／１２４１８１

(32)優先日　平成10年７月28日(1998.7.28)

(33)優先権主張国　米国（ＵＳ）

(71)出願人　591064003
サン・マイクロシステムズ・インコーポレーテッド
SUN MICROSYSTEMS, INCORPORATED
アメリカ合衆国　94303　カリフォルニア州・パロ　アルト・サン　アントニオ　ロード・901

(74)代理人　100096817
弁理士　五十嵐　孝雄　（外２名）

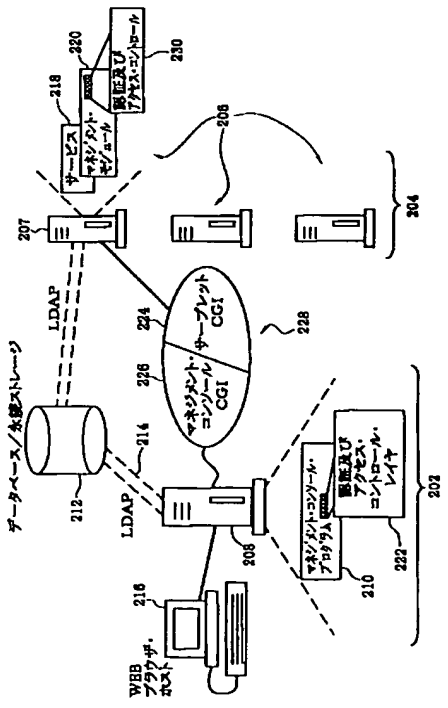
最終頁に続く

(54)【発明の名称】　コンピュータ・ネットワーク内のサービスを管理するためのマネジメント・コンソール・プログラム内の認証及びアクセス・コントロール

(57)【要約】　（修正有）

【課題】アドミニストレーション・サーバ・コンピュータから複数のサービス・ホスト・コンピュータ上に存在するサービス・マネージャへのアクセスを保護する。

【解決手段】ユーザ識別子及びパスワードを永続メモリ内に格納されたユーザ識別子及び対応するパスワードのリストと比較することにより、サービス・マネージャはユーザを認証する。システム・アドミニストレータがアドミニストレイティブ・アクセスを有するサービスのリストは、永続メモリ内のデータから導出する。システム・アドミニストレータがサービスのリストに含まれる１つ以上のサービスを管理するリクエストを行った際、永続メモリ内のアクセス・コントロール・データの調査により、アドミニストレータのアクセス・コントロールを、リクエストされたサービスを有するサービス・ホスト・コンピュータ上でベリファイする。



【特許請求の範囲】

【請求項１】　１つ以上のサービス・ホスト・コンピュータに対して接続されているアドミニストレーション・サーバ・コンピュータから前記１つ以上のサービス・ホスト・コンピュータ上に存在する複数の異なるサービスのアドミニストレーションへのアクセスを保護する方法であって、サービス・マネージャが前記アドミニストレーション・サーバ・コンピュータ上に存在する方法において、

選択されたユーザ識別子及び対応するプライベート・キーワードを提供する工程と、前記ユーザ識別子は前記複数の異なるサービスのうちの少なくとも１つへのアドミニストレイティブ・アクセスを有するユーザを識別するために設けられていることと、

前記サービス・マネージャのコントロール下において、前記選択されたユーザ識別子及び対応するプライベート・キーワードを永続ストレージ領域内に格納された複数のユーザ識別子及びプライベート・キーワードと比較することにより、ユーザを認証する工程と、

前記ユーザ識別子に関連するユーザがアドミニストレイティブ・アクセスを有するサービスのリストを導出する工程と、

前記導出したサービスのリストに含まれる前記複数のサービスのうちの選択された１つを管理するリクエストがあった際、前記永続ストレージ領域内の前記選択されたユーザ識別子に関連するアクセス・コントロール・データを調べることにより、前記選択されたユーザ識別子に関連するユーザが前記選択されたサービスへのアクセスを許可されていることを、前記選択されたサービスに関連するサービス・ホスト・コンピュータにおいてベリファイする工程と、

前記サービス・ホスト・コンピュータ上の１つ以上のマネジメント・ファイルをアドミニストレーション・サーバへ転送し、これによって、前記サービス・マネージャを使用したマネジメント・ファイルの操作を促進する工程とを含む方法。

【請求項２】　前記アドミニストレーション・サーバ・コンピュータは、ブラウザ・プログラムの実行に適したアドミニストレーション・クライアント・コンピュータに対して接続されており、前記選択されたユーザ識別子及び対応するプライベート・キーワードは、前記アドミニストレーション・クライアント・コンピュータ及びアドミニストレーション・サーバ・コンピュータの間の通信コネクションを通じて提供され、前記アドミニストレーション・サーバ・コンピュータ、前記アドミニストレーション・クライアント・コンピュータ及び前記１つ以上のサービス・ホスト・コンピュータの間の通信接続はインターネット・プロトコルを使用する請求項１に記載の方法。

【請求項３】　前記選択されたユーザ識別子及び対応するプライベート・キーワードを提供する工程は、前記アドミニストレーション・クライアント・コンピュータを通じて前記サービス・マネージャへログオンする工程を更に含む請求項１に記載の方法。

【請求項４】　前記ユーザを認証する工程は、前記ユーザ識別子及び対応するプライベート・キーワードを前記永続ストレージ領域へ伝達するために、ライトウェイト・ディレクトリ・アクセス・プロトコルを使用する工程を含む請求項１に記載の方法。

【請求項５】　前記各ユーザ識別子は対応するユーザ・プロフィールを有し、前記ユーザ・プロフィールは特定のサービス・マネージャ・ユーザに対応するグローバル・ユーザ・アイデンティティを表す請求項１に記載の方法。

【請求項６】　前記サービスのリストを導出する工程は、前記永続ストレージ領域を検索する工程を更に含み、前記永続ストレージ領域はユーザ・プロフィール・データベースを有し、前記ユーザ・プロフィール・データベースはユーザ・アクセス・レベル、許可可能なサービスのリスト及びパスワードをユーザ毎に含む請求項１に記載の方法。

【請求項７】　前記選択されたユーザ識別子に関連するユーザが前記サービスのリストから選択されたサービスへのアクセスを許可されていることを、前記サービス・ホスト・コンピュータにおいてベリファイする工程は、コモン・ゲートウェイ・インターフェースを使用することによって、前記選択されたユーザ識別子及び対応するプライベート・キーワードをホスト・サーバ・コンピュータへ伝達する工程を更に含む請求項１に記載の方法。

【請求項８】　前記サービス・ホスト・コンピュータは認証及びアクセス・コントロール・セグメントを有する請求項１に記載の方法。

【請求項９】　前記選択されたユーザ識別子及び対応するプライベート・キーワードは使用を目的として前記１つ以上のサービス・ホスト・コンピュータへ自動的に伝達される請求項１に記載の方法。

【請求項１０】　前記サービスのリストを、前記アドミニストレーション・クライアント・コンピュータ上に表示されたユーザ・インターフェース内で表示する工程を更に含む請求項１に記載の方法。

【請求項１１】　ホスト・サーバ・コンピュータ上のサービスの位置を決定するために、前記マネジメント・コンソール・プログラムによってサービス・ロケータを作成する工程を更に含む請求項１に記載の方法。

【請求項１２】　前記ホスト・サーバ上の１つ以上のマネジメント・ファイルをアドミニストレーション・サーバへ転送する工程は、前記アドミニストレーション・サーバ・コンピュータ上のコモン・ゲートウェイ・インターフェースを開始し、これによって、前記１つ以上のマネジメント・ファイル及び複数のオペレーティング・シ

ステム・コマンドの転送を可能にする工程を更に含む請求項１に記載の方法。

【請求項１３】　アドミニストレーション・サーバ・コンピュータから１つ以上のサービス・ホスト・コンピュータ上に存在するサービスのアドミニストレーションを保護するシステムであって、前記アドミニストレーション・サーバ・コンピュータが、ブラウザ型プログラムを有するアドミニストレーション・クライアントと、前記１つ以上のサービス・ホスト・コンピュータとに対してインターネット・プロトコルを使用して接続されているシステムにおいて、

ユーザ特権に関連するデータを格納するためのユーザ・プロフィール・データ・リポジトリと、前記データはユーザ・アクセス・レベル、サービスのリスト及びパスワードをユーザ毎に含むことと、

ユーザ識別子及び対応するキーワードを受信し、かつ前記ユーザ識別子及び対応するキーワードを前記ユーザ・プロフィール・データ・リポジトリへ送信するために、前記アドミニストレーション・サーバ・コンピュータ上に存在する通信インターフェースのサービス・マネージャ・サブコンポーネントと、前記１つ以上のサービス・ホスト上へ配置することに適したコンポーネント・コンフィギュレーション・ディレクトリと、前記コンポーネント・コンフィギュレーション・ディレクトリは前記複数のサービスに関連したマネジメント・モジュールを格納するためのコンポーネント・コンフィギュレーション・ファイルを含み、前記マネジメント・モジュールは前記複数のサービスを管理するために使用するマネジメント・データを含むことと、

前記ユーザ・プロフィール・データ・リポジトリ内に格納されたユーザ特権に関連するデータを調べることによってベリフィケーションを行うべく、前記ユーザ識別子及び対応するキーワードを受信し、かつ前記ユーザ識別子及び対応するキーワードを前記複数のサービス・ホスト・コンピュータへ送信するために、前記アドミニストレーション・サーバ・コンピュータ上に存在する前記通信インターフェースのサービス・ホスト・サブコンポーネントとを含むシステム。

【請求項１４】　１つ以上のサービス・ホスト・コンピュータと、アドミニストレーション・クライアント・コンピュータとへ接続されたアドミニストレーション・サーバ・コンピュータから前記１つ以上のサービス・ホスト・コンピュータ上に存在する複数の異なるサービスのアドミニストレーションへのアクセスを保護するシステムであって、サービス・マネージャが前記アドミニストレーション・サーバ・コンピュータ上に存在するシステムにおいて、

選択されたユーザ識別子及び対応するプライベート・キーワードを前記サービス・マネージャへ提供するために使用可能な前記アドミニストレーション・クライアント・コンピュータ及びアドミニストレーション・サーバ・コンピュータの間の通信コネクションと、前記ユーザ識別子は前記複数のサービスのうちの少なくとも１つに対するアドミニストレイティブ・アクセスを有するユーザを識別するために設けられていることと、

前記サービス・マネージャのコントロール下において、前記選択されたユーザ識別子及び対応するプライベート・キーワードを永続ストレージ領域内に格納された複数のユーザ識別子及びプライベート・キーワードと比較することにより、ユーザを認証するために形成されたオーセンティケータと、

前記ユーザ識別子に関連するユーザがアドミニストレイティブ・アクセスを有するサービスのリストを導出するためのアクセス・コントロール・メカニズムと、

前記導出したサービスのリストに含まれる前記複数のサービスのうちの選択された１つへアクセスすることを前記選択されたユーザ識別子に関連するユーザが許可されていることをベリファイするためのサービス・ホスト・ベリファイヤと、前記ベリファイヤは前記選択されたサービスに関連するサービス・ホスト・コンピュータ上に存在し、かつ前記永続ストレージ領域内の前記選択されたユーザ識別子に関連するアクセス・コントロール・データを使用することと、

前記サービス・ホスト・コンピュータ上の１つ以上のマネジメント・ファイルを前記アドミニストレーション・サーバ・コンピュータへ転送し、これによって、前記サービス・マネージャを使用したマネジメント・ファイルの操作を促進するためのデータ転送コンポーネントとを含むシステム。

【請求項１５】　１つ以上のサービス・ホスト・コンピュータへ接続されたアドミニストレーション・サーバ・コンピュータから前記１つ以上のサービス・ホスト・コンピュータ上に存在する複数の異なるサービスのアドミニストレーションへのアクセスを保護するコンピュータ・プログラミング命令を格納するために設けられたコンピュータ読み取り可能媒体であって、サービス・マネージャが前記アドミニストレーション・サーバ・コンピュータ上に存在するコンピュータ読み取り可能媒体において、

選択されたユーザ識別子及び対応するプライベート・キーワードを提供するためのコンピュータ・プログラミング命令と、前記ユーザ識別子は前記複数の異なるサービスのうちの少なくとも１つへのアドミニストレイティブ・アクセスを有するユーザを識別するために設けられていることと、

前記サービス・マネージャのコントロール下において、前記選択されたユーザ識別子及び対応するプライベート・キーワードを永続ストレージ領域内に格納された複数のユーザ識別子及びプライベート・キーワードと比較することにより、ユーザを認証するためのコンピュータ・

プログラミング命令と、
前記ユーザ識別子に関連するユーザがアドミニストレイティブ・アクセスを有するサービスのリストを導出するためのコンピュータ・プログラミング命令と、
前記導出したサービスのリストに含まれる前記複数のサービスのうちの選択された１つを管理するリクエストがあったとき、前記永続ストレージ領域内の前記選択されたユーザ識別子に関連するアクセス・コントロール・データを調べることにより、前記選択されたユーザ識別子に関連するユーザが前記選択されたサービスへのアクセスを許可されていることを、前記選択されたサービスに関連するサービス・ホスト・コンピュータにおいてベリファイするためのコンピュータ・プログラミング命令と、
前記サービス・ホスト・コンピュータ上の１つ以上のマネジメント・ファイルをアドミニストレーション・サーバへ転送し、これによって、前記サービス・マネージャを使用したマネジメント・ファイルの操作を促進するためのコンピュータ・プログラミング命令とを含むコンピュータ読み取り可能媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】一般的に、本発明はコンピュータ・ソフトウェア及びコンピュータ・ネットワーク・マネジメントに関する。より詳細には、本発明はサーバベースのマネジメント・ソフトウェアと、コンピュータ・ネットワーク内におけるソフトウェアの登録とに関する。

【０００２】

【従来の技術】近年、コンピュータ・ネットワークは、そのユーザ数またはサービス・エリアなどの規模の点で成長したのみならず、単一のネットワークが提供し、かつサポートできるサービス及びプロトコルの種類の点でも成長している。ニュース・サービスを読んだり、インターネットへアクセスしたりなど、エンドユーザが全ての種類のサービスへアクセスすることを、多くのコンピュータ・ネットワークは可能にしている。更に、これらのコンピュータ・ネットワークはユーザを１つの強制的なネットワーク・コミュニケーション・プロトコル、即ち、所定のネットワーク・コミュニケーション・プロトコルに束縛しない。コンピュータ・ネットワーク上で利用可能なサービスの急増により、これらのサービスを管理するシステム・アドミニストレータまたはネットワーク・アドミニストレータの負担が増大している。現在、一般的に、システム・アドミニストレータはソフトウェアを幾つかのサーバへインストールし、かつ管理する必要がある。一般的に、これらのサーバは１つ以上のサービスをそれぞれホストしている、即ち、１つ以上のサービスをネットワーク・ユーザへそれぞれ提供している。ネットワークの規模及びサービスの数次第では、これらのサービスの背後におけるソフトウェアのインストーリング、アップグレーディング及びトラブルシューティングなどの日々のマネジメントは、システム・アドミニストレータにとって単調で退屈なうえに間違いを起こしやすく、かつ長時間に及ぶタスクとなり得る。ネットワーク、サーバまたはサーバ・コンフィギュレーションを熟知していないシステム・アドミニストレータにとって、これは特に事実と言える。

【０００３】前記のように多くの種類のサービス及びアプリケーションを提供する大規模コンピュータ・ネットワークでは、エンドユーザ、即ち、クライアントがアクセスできる幾つかまたは多数のサーバ・マシンが一般的に存在する。一般的に、複数のサーバがネットワーク上に存在するという事実は、ネットワークの物理的コンフィギュレーションに通常は関心のない一般エンドユーザにとってトランスペアレントである（意識されるものでない）。コンピュータ・ネットワークの管理責務を有するシステム・アドミニストレータは、Ｗｅｂ（ウェブ）サーバなどのアドミニストレーション・サーバと一般的に称されるサーバ及びコンソールからコンピュータ・ネットワークを通常は管理する。図１はエンドユーザがアクセス可能であって、かつアドミニストレーション・サーバ（本発明の自動管理能力は備えていない）へ接続された複数のサーバを有するコンピュータ・ネットワークのブロック図である。Ｗｅｂサーバ、即ち、アドミニストレータ・サーバ１０６へ接続されたクライアント１０４として示すアドミニストレータ・コンソールを、コンピュータ・ネットワーク１０２は有する。複数の「サービス」サーバ１０８がＷｅｂサーバ１０６へ接続されている。アドミニストレーション・サーバ１０６から見た場合、複数のサーバ１０８はマネジメント・クライアントである。しかし、エンドユーザから見た場合、複数のサーバ１０８は、特定の機能をそれぞれ有するか、または特定のサービスをそれぞれ提供する単なるサーバにしか過ぎない。

【０００４】複数のサーバ１０８のうちの１つが有するアプリケーション・ソフトウェアに対して、アップデート、インストレーションまたは任意の種類のメンテナンスが実行されるとき、または新たなサーバがネットワーク１０２へ追加されるとき、システム・アドミニストレータはアドミニストレーション・サーバ１０６上のソフトウェアを適切に修正する必要がある。例えば、新たな機能が既存のメール・サーバへインストールされるか、または新たなメール・サーバが追加される場合、アドミニストレータはアップデートの時点における新たな機能またはサーバのロケーション及び他の情報を書き留める必要がある、即ち、覚えておく必要がある。アドミニストレータは新たなアプリケーションをサーバ１１０へインストールする。新たなアプリケーションの全てのマネジメント・モジュールのロケーション（ユニフォーム・

リソース・ロケータの形態をなし得る）を含むこの情報は、コンソール１０４にて入力される必要がある。アドミニストレータ・コンソール１０４においてマニュアルで入力された後、新たなソフトウェアまたはサーバを管理するために必要なこの情報はアドミニストレータ・サーバ１０６に反映される。この段階において、サーバ１１０上の全てのマネジメント・モジュールのロケーションはアドミニストレータ・コンソール１０４からシステム・アドミニストレータへ提供される。前記の例における新たなメール機能がアドミニストレータ・サーバ１０６へ「登録」されるまでは、エンドユーザはこの新たなメール機能を管理または適切に設定できない。エンドユーザによるソフトウェアの使用が可能になる前に、アドミニストレーション・サーバ１０６はマネジメント・クライアント１０８上の新たなメール機能に関連したマネジメント・モジュールを見つけるためのロケーションを知る必要がある。

【０００５】これはアドミニストレータにとって非効率的なプロセスであるとともに、ネットワーク上の新たなアプリケーションができるだけ早く利用可能になることを期待するエンドユーザにとって不便である。更に、後からアドミニストレータ・コンソールで入力する必要のある新たな機能またはサーバに関する情報をインストレーション中に書き留めるなどといったマニュアル・タスク、即ち、非自動タスクを、アドミニストレータは遂行する必要があるため、このプロセスはエラーを起こしやすい。適時に正しくインストールする必要のある頻繁なアップデート、コレクションまたは新たなバージョンを有する多数のアプリケーション（例：３０個は珍しくない）をそれぞれ有する多数のサーバが存在する場合、この問題は更に深刻である。この種のセッティングでは、ネットワーク・サービスの管理は非効率的であって長い時間を要し、かつエラーを起こしやすいのみならず、非実用的である。

【０００６】複数のサービス・ホストを有するこのＷｅｂサーバベースのネットワークにおける１つの問題点としては、ユーザ認証メカニズムの設計及び実装が挙げられる。ユーザが許可されたオペレーションの実行または許可されたファイルへのアクセスのみを行うことを保証する認証プロトコルまたは認証メカニズムを、Ｗｅｂサーバベースのコンピュータ・ネットワークまたは任意の種類のコンピュータ・ネットワークは有する必要がある。複数のサービス・ホスト上のサービスを管理するケースでは、これらのホスト上におけるサービスを維持する責務を有する１人以上のシステム・アドミニストレータが存在し得る。全ての可能なオペレーションをＷｅｂサーバ及びサービス・ホスト上で実行するための完全な権限付与を特定のアドミニストレータに与えずに、この完全な権限付与を例えばシニア・システム・アドミニストレータ、即ち、「スーパー」システム・アドミニストレータに対してのみ与え得る。従って、複数のホスト上におけるサービスの管理は、アドミニストレーション・インターフェースを通じて行われるアドミニストレーション・タスクであるため、ある種のユーザ認証が必要である。

【０００７】認証はＷｅｂベースのネットワークに対して存在するが、従来のユーザ権限付与のためのインプリメンテーション及びデザインは非能率的であり、かつくどい。ここでいう認証とは、アドミニストレーション・コンソール上のブラウザからネットワーク内のサービス・ホスト上のサービスを管理するための、システム・アドミニストレータまたはネットワーク・アドミニストレータのベリフィケーション及び権限付与である。一般的に、サービス・ホスト上の各サービスと、その１つ以上のマネジメント・モジュールとは異なる認証メカニズム、即ち、スタンダードを有する。認証及びアクセス・コントロールをＷｅｂベースのシステム上で分散形式でインプリメントするためのプロトコルまたはプロセスに関する明らかなスタンダードは存在しない。複数のサービス・ホストは互いに通信していないため、システム・アドミニストレータがサービス・ホストへサインオンするごとに、このシステム・アドミニストレータは再認証を行う必要がある。ブラウザ・プログラムは、任意の種類のオペレーティング・システムを実行しているクライアント上で実行可能である。従って、アドミニストレータが使用するブラウザはＵＮＩＸ（ユニックス）ベースのクライアント上に存在せず、かつ既知のＵＮＩＸアイデンティティを有さないことがある。ブラウザが既知のＵＮＩＸアイデンティティを有さないことにより、アイデンティティを１つのサービス・ホストから別のサービス・ホストへ伝達できない。従って、システム・アドミニストレータは単一のアイデンティティ、即ち、グローバルに承認されたアイデンティティを有していないため、システム・アドミニストレータは各サービス・ホストの認証プロセスを通過する必要がある。

【０００８】従って、インストレーション中にセントラル・ロケーションで自動的に登録され、かつ周知のロケーションからアクセス可能なアプリケーション及びサービスを管理するために必要な任意のソフトウェアを有することにより、コンピュータ・ネットワーク上で利用できるエンドユーザ・アプリケーション・ソフトウェア及びサービスをセントラル・ロケーションから管理することが望ましい。更に、Ｗｅｂサーバの環境内のファンクションと、ユーザ・アイデンティティ及びアクセス・コントロールのためのサーバの既存のシステムとへのシングル・サインオンを提供する認証メカニズムを有することが望ましい。更に、Ｗｅｂサーバベースのネットワーク内のブラウザからサービスを管理するユーザに対し、ユニバーサル・アイデンティティを割り当てることにより、これをセントラル・ロケーションから達成すること

が望ましい。

【0009】

【発明の概要】本発明の前記の目的を達成すべく、アドミニストレーション・サーバ・コンピュータから1つ以上のサービス・ホスト・コンピュータ上に存在するサービスのアドミニストレーションのためのサービス・マネージャへのアクセスを保護する方法を開示する。本発明の好ましい実施形態では、ユーザ名などのユーザ識別子と、対応するパスワードとをサービス・マネージャへ提供する。ユーザ識別子はサービスへのアドミニストレイティブ・アクセスを有するシステム・アドミニストレータに関連している。ユーザ識別子及びパスワードを永続メモリ内に格納されたユーザ識別子及び対応するパスワードのリストと比較することにより、サービス・マネージャはユーザを認証する。システム・アドミニストレータがアドミニストレイティブ・アクセスを有するサービスのリストは、永続メモリ内のデータから導出する。システム・アドミニストレータがサービスのリストに含まれる1つ以上のサービスを管理するリクエストを行った際、永続メモリ内のアクセス・コントロール・データを調べることにより、アドミニストレータのアクセス・コントロールを、前記のリクエストされたサービスを有するサービス・ホスト・コンピュータ上でベリファイする。マネジメント・ファイルをサービス・ホスト・コンピュータからアドミニストレーション・サーバ・コンピュータへ転送し、これによって、サービス・マネージャを使用したマネジメント・ファイルの操作を促進する。

【0010】別の好ましい実施形態では、アドミニストレーション・サーバ・コンピュータはＷｅｂブラウザなどのブラウザ・プログラムを実行するアドミニストレーション・クライアント・コンピュータへ接続されている。ユーザ識別子及びパスワードを、アドミニストレーション・クライアント・コンピュータ及びアドミニストレーション・サーバ・コンピュータの間の通信コネクションを通じてアドミニストレーション・サーバ・コンピュータへ提供する。アドミニストレーション・サーバ・コンピュータ及びアドミニストレーション・クライアント・コンピュータの間の通信コネクションと、アドミニストレーション・サーバ・コンピュータ及びサービス・ホスト・コンピュータの間の通信コネクションとは、ＴＣＰ／ＩＰなどのインターネット・プロトコルを使用している。

【0011】本発明の別の態様では、コンピュータ・ネットワーク内のホスト・サービス・コンピュータ上に存在するサービスのアドミニストレーションのためのサービス・マネージャへのアクセスを保護するシステムを開示する。好ましい実施形態では、サービス・マネージャは複数のホスト・サービス・コンピュータへ接続されたアドミニストレーション・サーバ・コンピュータ上に存在し、かつアドミニストレーション・クライアント・コンピュータへ接続されている。アドミニストレーション・クライアント・コンピュータ及びアドミニストレーション・サーバ・コンピュータの間の通信コネクションは、ユーザ識別子及びパスワードをサービス・マネージャへ提供するために使用される。ユーザ識別子は少なくとも1つのサービスへのアドミニストレイティブ・アクセスを有するユーザ、一般的には、システム・アドミニストレータを表す。サービス・マネージャのコントロール下において、オーセンティケータはユーザ識別子及びパスワードを永続メモリ内に格納されたユーザ識別子及びパスワードのリストと比較する。識別子及びパスワードに関連するシステム・アドミニストレータがアドミニストレイティブ・アクセスを有するサービスのリストを、アクセス・コントロール・メカニズムは導出する。サービス・ホスト・コンピュータ上に存在するサービス・ホスト・ベリファイヤは、永続メモリ内に格納されているシステム・アドミニストレータに関連するアクセス・コントロール・データを使用することにより、システム・アドミニストレータがサービスのリストから選択されたサービスへのアクセスを許可されていることをベリファイする。データ転送コンポーネントは、サービス・ホスト・コンピュータ上に存在するマネジメント・ファイルをアドミニストレーション・サーバ・コンピュータへ転送し、これによって、サービス・マネージャを使用したマネジメント・ファイルの操作を促進する。

【0012】本発明の別の態様では、ブラウザ・プログラムを有するアドミニストレーション・クライアントと、サービス・ホスト・コンピュータとへＴＣＰ／ＩＰなどのインターネット・プロトコルを使用して接続されたアドミニストレーション・サーバからコンピュータ・ネットワーク内のサービス・ホスト・コンピュータ上に存在するサービスのアドミニストレーションを保護するシステムを開示する。好ましい実施形態では、ユーザ・プロフィール・データ・リポジトリはユーザ・アクセス・レベル、サービスのリスト及びパスワードを含むユーザ特権に関するデータを格納する。アドミニストレーション・サーバ上に存在するサービス・マネージャ・サブコンポーネントを有する通信インターフェースは、ユーザ名及びパスワードを受信し、これらの情報をユーザ・プロフィール・データ・リポジトリへ送信する。サービス・ホスト上に存在可能なコンポーネント・コンフィギュレーション・ディレクトリは、サービスに属するマネジメント・モジュールを格納するコンポーネント・コンフィギュレーション・ファイルを含む。マネジメント・モジュールはサービスのアドミニストレーションに使用可能なマネジメント・データを含む。更に、通信インターフェースはアドミニストレーション・サーバ・コンピュータ上に存在するサービス・ホスト・サブコンポーネントを有する。サービス・ホスト・サブコンポーネントはユーザ名及びパスワードを受信し、サービス・ホスト

上でのベリフィケーションのために、これらの情報をサービス・ホストへ送信し、ベリフィケーションはユーザ・プロフィール・データ・リポジトリ内に格納されているユーザ特権に関するデータを調べることによって行われる。

【００１３】本発明とその更なる効果は添付図面に基づく以下の説明から最もよく理解できる。

【００１４】

【発明の実施の形態】本発明の好ましい実施形態を詳述する。好ましい実施形態の例を添付図面に示す。本発明を好ましい実施形態に関連して詳述するが、これは本発明を１つの好ましい実施形態に限定することを意図するものではない。逆に、請求の範囲に定める本発明の趣旨及び範囲に含まれる別例、変更例及び等価なものを包含することを意図している。

【００１５】コンピュータ・ネットワーク内のセントラル・ロケーションからソフトウェア・アプリケーション及びサービスを管理するための方法及びシステムを複数の図面に示す。複数のサーバと、大きなエンドユーザ・ベースとを有する大規模コンピュータ・ネットワークでは、ネットワーク上におけるアプリケーション及びソフトウェアの管理は多くの時間を要するエラーを起こしやすいタスクである。一般的に、システム・アドミニストレータは新たなアプリケーション、即ち、サービスをサービス・ホスト、即ち、複数のネットワーク・サーバのうちの１つへインストールする。このインストールは、通常、インストール先のサーバにおいて行われる。システム・アドミニストレータはアプリケーションの管理に関する情報（特に、マネジメント・モジュールのファイルのロケーション及び名前）をマニュアルで書き留める。次いで、この情報をアドミニストレータ・コンソールを通じてアドミニストレータ・サーバへ入力する。アドミニストレータ・サーバ（例：Ｗｅｂサーバ）が新たなアプリケーション・マネジメント・モジュールのロケーションを認識した後、エンドユーザは新たなアプリケーションへアクセス可能である。頻繁なアップデート、変更または置換を必要とする多数のアプリケーションを有する多数のサーバがネットワーク上に存在する際、このプロセスは重荷であり、かつ非効率的である。アプリケーションの受信後、このアプリケーションが即座に利用可能になることへの期待が大きい場合、この問題はエンドユーザにとって深刻である。前記の非自動の２工程プロセスは、ネットワーク上のユーザがアプリケーションを利用できるようになるまでの時間を増大させる。

【００１６】本発明は新たなアプリケーション及びサービスをＷｅｂサーバなどのセントラル・マネジメント・ロケーションにおいて登録するプロセスを自動化する方法であり、これによって、システム・アドミニストレータが記憶する必要のある情報量を低減し、かつエンドユーザがサービスを更に早く利用できるようにする。本実施形態において、本発明はネットワーク上の他のサーバ、即ち、サービス・ホストを管理するアドミニストレーション・サーバ上に存在するマネジメント・コンソール・プログラムを持つことを含む。そして、前記の他のサーバはアドミニストレーション・サーバの「クライアント」である点からマネジメント・クライアントとも呼ばれる。更に、本実施形態はマネジメント情報を格納するためのデータベースを含む永続ストレージ領域を含み、かつサービス・ホスト上のサービスと、各マネジメント・クライアントに関連する「周知の」ディレクトリとに関する認証情報を使用する（例：システム・アドミニストレータまたはネットワーク・アドミニストレータ）。以下に詳述する別の好ましい実施形態では、例えば、ストレージ領域を１つのサーバのみへ関連させる代わりに、このストレージ領域をネットワーク上へ分散させ得る。別の好ましい実施形態では、マネジメント・コンソール・プログラム全体をアドミニストレーション・サーバへインストールせずに、このマネジメント・コンソール・プログラムをサーバ及びアドミニストレータ・クライアント・マシンの間に分散させ得る。これらのコンポーネントは図２に示す。

【００１７】図２は本発明の１つの実施形態に基づくコンピュータ・ネットワークのサーバ側のコンポーネントのブロック図である。ネットワーク全体（図示略）のサーバ側コンフィギュレーション２００は、２つのセクション、即ち、アドミニストレーション側を表すセクション２０２と、ネットワーク・サーバ、即ち、サービス・ホストを表すセクション２０４とを有するものと見なし得る。サービスの提供、アプリケーションの実行または他のネットワーク・オペレーションの実施のために、ネットワーク・サーバ２０６へのアクセスが可能なクライアント・マシン上のネットワーク・エンドユーザは図２に示されていない。コンピュータ・ネットワークのエンドユーザがネットワーク上のサービス及びアプリケーションを更に早く利用可能となり、かつ、これらのサービス及びアプリケーションが頻繁にダウンしなくなった場合、コンピュータ・ネットワークのエンドユーザは本発明の受益者の一人となる。しかし、本実施形態において、本発明はシステム・アドミニストレータまたはネットワーク・マネージャ（即ち、ユーザ）によって使用される。

【００１８】本実施形態では、マネジメント・クライアント２０６はＷｅｂサーバ２０８を通じて管理される。別の好ましい実施形態では、サーバ２０８は別の種類のサーバ（更に汎用的なアドミニストレーション・サーバなど）であるか、またはネットワークのサイズ及びサーバのキャパシティに基づく別の機能を有するサーバであり得る。いずれにせよ、ネットワーク内のサーバ２０８はマネジメント・クライアント２０６を管理する役割を有する。サーバ２０８の１つの特徴としては、以下に詳

述するマネジメント・コンソール・プログラム２１０を有する点が挙げられる。Ｗｅｂサーバ２０８の別の特徴としては、サービス・マネジメント・モジュール情報を格納する永続ストレージ領域データベース２１２へのアクセスを有する点が挙げられる。Ｗｅｂサーバ２０８はライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）２１４を通じてストレージ２１２と通信する。別の好ましい実施形態では、別のデータ・アクセス・プロトコルをサーバ２０８及びストレージ領域２１２の間で使用し得る。更に、ストレージ領域２１２はマネジメント・クライアント２０６からのアクセスが可能である。本実施形態において、永続ストレージ２１２はデータを階層フォーマットで格納する信頼性の高いデータベースである。別の好ましい実施形態において、データベースはリレーショナル・データベース・フォーマットを有し得る。即ち、データベースはデータをオブジェクト指向のデータ・リポジトリへ格納可能である。更に、別の好ましい実施形態では、ストレージ２１２は、マネジメント・クライアント２０６の永続ストレージ領域部分と、Ｗｅｂサーバ２０８の永続ストレージ領域部分と、ネットワークで利用可能であって、かつサーバによるアクセスが可能な他の永続ストレージ媒体の永続ストレージ領域部分とへ分散させ得る。

【００１９】前記のように、本発明はシステム・アドミニストレータが主に使用する。アドミニストレータは特別なクライアント・アドミニストレータ・コンソール２１６を通じてサーバ２０８へアクセスする。本実施形態では、アドミニストレータがサーバ２０８へアクセスすることを可能にするＷｅｂベースのブラウザ・プログラムを、コンソール２１６は備えている。より具体的には、コンソール２１６はマネジメント・コンソール・プログラム２１０及びストレージ領域２１２を使用する。ブラウザ・ホスト２１６から見た場合、サーバ２０８はマネジメント・コンソール・ホストと呼ぶことができる。以下に詳述するように、マネジメント・クライアント２０６上のソフトウェア・アプリケーション及びサービスを管理するために、システム・アドミニストレータはブラウザ・ホスト２１６を使用できる。

【００２０】マネジメント・クライアント２０６はネットワーク上の全てのサーバまたは幾つかのサーバを含み得る。システム・アドミニストレータがＷｅｂサーバ２０８を通じて管理するサーバは、ＬＤＡＰを通じてストレージ２１２と通信する。各マネジメント・クライアントは符号２１８で示す１つ以上のサービスと、符号２２０で示す１つ以上の対応するマネジメント・モジュールとをサービス・ホスト２０７上に有する。新たなサービスをインストールする際、または既存のサービスをアップグレードする際、マネジメント・モジュール領域２２０内のエントリは変更される。以下に詳述するように、この変更は永続ストレージ２１２内の対応するエントリ内へ反映される。図２では、サービス２１８をマネジメント・モジュール２２０から分離して示しているが、これら２つのコンポーネントは互いに必要不可欠である。換言するならば、サービスのマネジメント・モジュールは、サービスの主要部、即ち、機能モジュールと一体に結合している。しかし、２つのコンポーネントは別々の役割を依然有している。マネジメント・モジュール２２０はコンフィギュレーション・ファイル内に格納される。コンフィギュレーション・コンポーネント・ディレクトリは以下に詳述する。別の好ましい実施形態において、マネジメント・モジュール２２０内の情報は別の非マネジメント・ファイルを含むデータベースまたはスタンダード・ディレクトリなどの別のフォーマットで格納可能である。

【００２１】マネジメント・コンソール・プログラムに関連する図２の残りのコンポーネントは認証及びアクセス・コントロールの機能を管理する。マネジメント・コンソール・プログラム２１０は以下に図８～図１１に関連して詳述するユーザ・ベリフィケーション機能及びユーザ認証機能を実施する認証レイヤ２２２を有する。プログラムを実行するためにＷｅｂサーバが使用するコモン・ゲートウェイ・インターフェース・プログラム、即ち、ＣＧＩプログラムがコンソール・ホスト２０８に関連している。本実施形態では、ＣＧＩプログラム２２４はプログラムをコンソール・ホスト２０８から実行するために使用され、かつ２つの部分、即ち、マネジメント・コンソールＣＧＩ２２６及びサーブレットＣＧＩ２２８へ論理的に分割されている。マネジメント・コンソールＣＧＩ２２６はマネジメント・コンソール・プログラム２０８と通信する。更に、マネジメント・コンソールＣＧＩ２２６は図９及び図１０に関連して以下に詳述する。サーブレットＣＧＩ２２８は認証データをコンソール・ホスト２０８からサービス・ホスト２０６へ伝達する。更に、サーブレットＣＧＩ２２８は当該技術分野において周知のコンポーネントである。

【００２２】マネジメント・モジュール・コンポーネント２２０の一部である対応する認証及びアクセス・コントロール・レイヤ２３０がサービス・ホスト２０６上に存在する。認証レイヤ２３０はデータをコンソール・ホスト２０８からサーブレットＣＧＩ２２８を通じて受信する。マネジメント・コンソール・プログラムを特定のサービスの管理に使用すべくログオンするシステム・アドミニストレータに対する前記の特定のサービスの管理権限付与を保証するために、これらのコンポーネントは使用される。更に、これらのコンポーネントは「スーパー」システム・アドミニストレータがアドミニストレータと、マネジメント・コンソール・フレームワーク内の特定の権限とを追加及び削除することを可能にする。本実施形態では、この機能性は図８に示すグラフィカル・ユーザ・インターフェースを通じて明らかにする。サー

ビス・ホスト２０６は永続データ・ストレージ２１２を用いてユーザのアクセス・コントロール及び権限付与を再認証する。

【００２３】図３は本発明の１つの実施形態に基づく新たなサービスをネットワーク上で登録するプロセスの概要を示すフローチャートである。新たなサービスの登録、サービスのアップグレード、または新たなマネジメント・クライアントの追加をネットワークに対して行う際、システム・アドミニストレータが実施するステップをフローチャートは示す。ステップ３０２では、サービスを特定のマネジメント・クライアントへインストールする。これはブラウザ・ホストとして機能するクライアント・マシンを通じて一般的に行われ、通常は、システム・アドミニストレータがこれを実施する。サービスに関連するマネジメント・モジュールはマネジメント・クライアント上にもインストールされている実行可能なコードのセグメントである。メール・サーバ上のマネジメント・モジュールの例としては、１エンドユーザあたりの最大割当量、即ち、ユーザが占有できるメモリの最大量を示すモジュールが挙げられる。別の例としては、顧客のＷｅｂサイトをホストするＩＳＰ（インターネット・サービス・プロバイダ）が所有するＷｅｂサーバが挙げられる。この場合、マネジメント・モジュールはＷｅｂサーバ上での新たなＷｅｂサイトの追加を管理し得る。

【００２４】マネジメント・モジュールは幾種類かあるマネジメント・モジュールのうちの１つであり得る。本実施形態では、これらのマネジメント・モジュールの種類としては、ブラウザベースのマネジメント・モジュール、Ｘベースのマネジメント・モジュール及びコマンド・ライン・マネジメント・モジュールが挙げられる。ブラウザベースのマネジメント・モジュールはＷｅｂブラウザ内で実行されるアプリケーションに関連している。アプリケーションの種類の大半がＷｅｂブラウザ内で実行されるアプリケーションであることを予想できる。Ｘベースのマネジメント・モジュールは、ＵＮＩＸオペレーティング・システムのコンポーネントであるＸプロトコルに基づいて実行されるスタンドアロン・アプリケーションに一般的に関連している。一般的に、これらのアプリケーションはブラウザ内から実行されることはなく、オペレーティング・システム・シェルから実行される。これはＵＮＩＸベースのグラフィカル・ユーザ・インターフェースである周知のスタンダードＸウィンドウズに由来する。コマンド・ライン・マネジメント・モジュールはコマンド・ラインを用いて管理するアプリケーションに関連している。しかし、コマンド・ライン・マネジメント・モジュールを埋め込み、かつこれをＷｅｂブラウザから実行できる。コマンド・ラインは以下に詳述するようにランタイム・パラメータを有していても良く、またランタイム・パラメータを有さなくても良い。コマンド・ライン・コマンドの例としては、「ls」（ファイルのリストを獲得する）、「whoami」（現在のユーザに関する情報を返す）及び「ｐｓ」（パフォーマンス・ステータスに関する情報を提供する）が挙げられる。別の好ましい実施形態では、別の種類のマネジメント・モジュールをインストールできる。

【００２５】ステップ３０４では、システム・アドミニストレータはサービス及びマネジメント・モジュールをマネジメント・クライアント上へ登録する。本実施形態では、mc_regと呼ばれるコマンドをマネジメント・クライアント上で実行することによって、この登録は行われる。サービス及びマネジメント・モジュールを登録することにより、インストールされたモジュールの種類がアドミニストレーション・サーバ（図２のサーバ２０８）へ通知される。一般的に、システム・アドミニストレータは幾つかの新たなサービスを複数の異なるマネジメント・クライアントへ登録する。従って、ステップ３０２及びステップ３０４を、複数の異なるマネジメント・クライアント上の幾つかのサービスに対して反復して実施することになる。サービスをサービス・ホストへ登録した後、マネジメント・データを格納するコンポーネント・コンフィギュレーション・ファイルと称される特定のファイルが形成され、かつサービス・ホスト上のコンポーネント・コンフィギュレーション・ディレクトリ内へ格納される。ステップ３０４を図４に基づいて以下に詳述する。

【００２６】ステップ３０６では、「ディスカバー」ルーチンをマネジメント・コンソール・プログラム２１０に関連するグラフィカル・ユーザ・インターフェースを通じて開始するとともに、この「ディスカバー」ルーチンをサービス・ホスト上で実行する。このルーチンはマネジメント・コンソール・プログラムによる特定のサービス・ホストの登録を可能にする。特定の１つのサービス・ホストまたはサービス・ホスト群へアクセスし、かつ何がそこに登録されているかをチェックする命令を、システム・アドミニストレータは例えばブラウザ・ホスト２１６を通じてマネジメント・コンソールへ与える。本実施形態では、システム・アドミニストレータが指し示すサービス・ホスト上のコンポーネント・コンフィギュレーション・ディレクトリと呼ばれる周知のディレクトリをマネジメント・コンソールがチェックすることによって、これは行われる。ステップ３０６は図５において更に詳述する。好ましい実施形態では、サービスがステップ３０２でインストールされた時点において、ディスカバー・ルーチンをサービス・ホスト上でローカルに実行できる。サービス・ホストは、このリモート・ディスカバー、即ち、オート・ディスカバーの結果をマネジメント・コンソール・プログラムへ一斉同報し得る。本実施形態では、システム・アドミニストレータが変更、アップグレードまたは追加を最近行った全てのサービス

・ホストを登録するために、システム・アドミニストレータはマネジメント・コンソール・プログラムへ命令し得る。本実施形態では、マネジメント・コンソール・プログラムはこれらのサービス・ホストのチェックを行い、かつコンポーネント・コンフィギュレーション・ディレクトリをチェックすることにより、任意のアップデートを登録する。変更された全てのサービス・ホストを登録した後、エンドユーザはサービス、即ち、アプリケーションの使用を開始可能であり、登録プロセスが完了する。

【００２７】図４は本発明の１つの実施形態に基づくサービスを登録する図３のステップ３０４の詳細を示すフローチャートである。ステップ３０４はシステム・アドミニストレータから命令を受けることにより、新たなサービスがサービス・ホストへ登録されたことをマネジメント・コンソールが後から認識できるように、新たなサービスをサービス・ホストへ登録するプロセスを提供している。ステップ４０２では、サービス、即ち、アプリケーションの種類がサービス・ホストに対して識別される。前記のように、本実施形態では、サービスは３つの種類、即ち、ブラウザベース、Ｘベース及びコマンド・ラインのうちの１つであり得る。別の好ましい実施形態では、更に別の種類のサービスを入力し得る。本実施形態では、このステップはサービス・ホスト上で実施される。そして、このステップはアプリケーションの種類をマネジメント・コンソールへ通知する１つの方法といえる。別の好ましい実施形態では、この情報はブラウザ・ホストにおいて入力可能である。ステップ４０２の後、サービス・ホストで入力する情報は、識別されたサービスの種類に基づいている。サービスがＷｅｂベースのサービスである場合、フローチャートはステップ４０４へ移行する。ステップ４０４では、システム・アドミニストレータはサービス・ホスト上に存在するサービスのマネジメント・モジュールのロケーションを入力する。Ｗｅｂベースのサービスの場合、このロケーションは一般的にユニフォーム・リソース・ロケータ、即ち、ＵＲＬの形態をなす。ステップ４０６では、サービスの種類と、マネジメント・モジュールのＵＲＬとをサービス・ホスト上の周知のロケーションへパラメータとして保存する。本実施形態では、これら２つのコンポーネントと呼ばれる情報アイテムは、コンポーネント・コンフィギュレーション・ディレクトリと呼ばれるディレクトリ内のコンポーネント・コンフィギュレーション・ファイルと呼ばれるＵＮＩＸファイルへ保存される。別の好ましい実施形態では、これらのコンポーネントを格納するためにサービス・ホスト上の別のディレクトリを使用し得る。

【００２８】ステップ４０８では、コンポーネント識別子をサービス・マネジメント・モジュールに含まれる２つのコンポーネントへそれぞれ割り当てる。本実施形態では、コンポーネント識別子は２つの部分、即ち、（１）固有識別子（ソラリス・パッケージ名（例：SUNWFTP）など）及び（２）バージョン番号からなる。従って、ＵＲＬコンポーネント及びサービスの種類コンポーネントは、コンポーネント識別子を割り当てられ、かつコンポーネント・コンフィギュレーション・ディレクトリ内のファイルへ保存される。更に、サービスに関する「ユーザフレンドリー」な名前を入力する（この時点まで、この名前は固有の長い不可解な名前である）。このユーザフレンドリーな名前は図６に基づいて以下に詳述するグラフィカル・ユーザ・インターフェース上に表示される名前である。ステップ４２０では、ステップ４０６及びステップ４０８で説明したデータ、即ち、コンポーネントをコンポーネント・コンフィギュレーション・ディレクトリ内の適切なファイル内へ格納する。従って、ステップ４２０の後、Ｗｅｂベースのサービスのための図３のステップ３０６（「ディスカバリ」プロセス）を実施するために必要な全ての情報は、周知のディレクトリに含まれる適切なファイル内へ格納され、プロセスが完了する。

【００２９】再びステップ４０２へ戻り、サービスの種類がＸベースのサービスである場合、コントロールはステップ４１０へ移行する。前記のように、Ｘベースのサービスは、ＵＮＩＸオペレーティング・システムのコンポーネントであるＸプロトコルに基づいて実行されるスタンドアロン・アプリケーションに一般的に関連している。ステップ４１０では、システム・アドミニストレータはＸベースのアプリケーションを呼び出すために必要なパスを入力する。ステップ４１２では、Ｘベースのアプリケーションを呼び出すために、ＵＮＩＸユーザ及びユーザ・グループを入力する。次いで、コントロールはステップ４０８へ移行する。ステップ４０８では、コンポーネント識別子をパス、ユーザ名及びグループへそれぞれ割り当てる。ステップ４２０では、コンポーネント識別子はコンポーネント・コンフィギュレーション・ディレクトリ内の適切なファイルへ格納される。

【００３０】コマンド・ライン・タイプのマネジメント・モジュールの場合、ステップ４１４に示すように、システム・アドミニストレータはＸベースのマネジメント・モジュールの場合のデータに類似したデータ、即ち、コマンド・ラインを呼び出すためのパスと、ＵＮＩＸアプリケーションを呼び出すために必要なＵＮＩＸユーザ及びグループの名前とを入力する。ステップ４１６では、システム・アドミニストレータはランタイム・パラメータがコマンド内に存在するか否かを決定する（コマンド・ライン・タイプのマネジメント・モジュールへ反映される）。これらのパラメータはサービスが登録された時点で入力されるのではなく、エンドユーザがコマンドを実行した時点で入力される。グラフィカル・ユーザ・インターフェースはエンドユーザがランタイム・パラ

メータ（例：サービスを使用する時点において、ユーザが選択できるオプション）を入力可能か否かを反映すべく変更、即ち、カスタマイズされている。ランタイム・パラメータが存在する場合、システム・アドミニストレータはこれらをマネジメント・コンソールのグラフィカル・ユーザ・インターフェースからのプロンプトに応じて供給する。ステップ４１８では、システム・アドミニストレータはコマンドが要求するスタティック・パラメータを入力する。コマンドがランタイム・パラメータを有するか否かとは無関係に、コマンド・ライン・タイプのマネジメント・モジュールはスタティック・パラメータを常に有する。次いで、コントロールはステップ４０８へ移行する。ステップ４０８では、Ｘベースのマネジメント・モジュール及びＷｅｂベースのマネジメント・モジュールで行ったのと同様に、コンポーネント識別子を全てのデータへ割り当てる。次いで、ステップ４２０では、コンポーネント識別子はコンフィギュレーション・コンポーネント・ディレクトリ内に格納されているファイルに保存される。本実施形態では、ファイル名は「コンポーネント識別子－バージョン番号」のフォーマットを有する。各コンポーネントが１つのファイルを有するディレクトリ内に登録されているコンポーネントの総数の決定を、この「コンポーネント識別子－バージョン番号」のフォーマットは促進する。別の好ましい実施形態では、１つのコマンドにつき１つのファイルを設けた別のフォーマット（例：コンポーネント識別子－コマンド・ナンバー）をファイル名は有することが可能である。

【００３１】図５は本発明の１つの実施形態に基づく図３のステップ３０６の詳細を示すフローチャートである。本実施形態では、サービス・ホスト上に存在するサービスの全てのマネジメント・モジュールを含む実行中のコンポーネント・ソフトウェア・セグメントを、サービス・ホストは有する。コンポーネント・コンフィギュレーション・ディレクトリはこのセグメント内に存在する。更に、アドミニストレーション・サーバ上に存在するマネジメント・コンソール・プログラムにも含まれるコードを含むマネジメント・コンソール・フレームワーク・セグメントを、サービス・ホストは有する。例えば、mc_regコマンドと、Ｘベースのマネジメント・プログラム及びコマンド・ライン・マネジメント・プログラムをリモートで実行するためのプログラムであるＩＳＰリモート・シェル・コードとは、マネジメント・コンソール及びサービス・ホストの両方に存在する。まだ登録されていないマネジメント・モジュールのサービス・ホスト上のコンポーネント・ソフトウェア・セグメントをマネジメント・コンソール・フレームワーク・セグメント内のソフトウェアを使用して探索するディスカバリ・プロセスを図５は示す。

【００３２】ステップ５０２では、システム・アドミニストレータはサービス・ホスト名またはサービス名をブラウザ・ホスト上のグラフィカル・ユーザ・インターフェースを通じて特定する。本実施形態で使用するグラフィカル・ユーザ・インターフェースの例は、図６、図７及び図８に詳細に示す。前記のように、利用可能な幾つかのサービスをそれぞれ有する多数のサービス・ホストが存在し得る。これらの選択はユーザ・インターフェースを通じてシステム・アドミニストレータへ提供される。一般的に、アドミニストレータは最近変更または追加されたサービスを含む全てのサービス・ホストを選択し、かつこれら全てのサービス・ホストをブラウザ・ホストから一度に入力する。ステップ５０４では、コンポーネント・コンフィギュレーション・ファイルを求めて周知のディレクトリをスキャンするために、マネジメント・コンソール・ホストはステップ５０２で特定された１つ以上のサービス・ホストへ接続する。本実施形態では、この周知のディレクトリはコンポーネント・コンフィギュレーション・ディレクトリである。Ｗｅｂベースのプログラムを Ｗｅｂサーバから起動するために一般的に使用され、かつ当該技術分野でよく知られたスタンダードＣＧＩ（コモン・ゲートウェイ・インターフェース）プログラムを通じて、マネジメント・コンソールはサービス・ホストと通信する。別の好ましい実施形態では、アドミニストレーション・サーバがＷｅｂベースのサーバでない場合、ＣＧＩプログラムは必要ない。前記のスキャンニングはコマンド・ライン・プログラムを使用して実行される。このコマンド・ライン・プログラムはコマンドをネットワーク・コネクションを通じて送信し、かつこのコマンドをデスティネーション・サーバ上で実行させる。より具体的には、本実施形態では、マネジメント・コンソールが前記のコマンドをネットワーク・コネクションを通じてサービス・ホスト上で実行する。本実施形態では、これはＩＳＰリモート・シェル・プロトコルを使用して行われる。従って、スキャンニング中、コンポーネント・コンフィギュレーション・ファイルのリストを獲得するために、ＵＮＩＸ「リスト・ファイル」コマンドであるlsをコンポーネント・コンフィギュレーション・ディレクトリ内で実行する。マネジメント・コンソールに登録する必要のあるファイルのリストはアドミニストレーション・サーバへ送信される。

【００３３】ステップ５０６では、マネジメント・コンソールはステップ５０２で特定された全てのサービス・ホスト上で「見つかった」ファイルのリストを調べる。次いで、これらのファイルの内容をリトリーブするために、マネジメント・コンソール及びサービス・ホストの間の同じコネクションを使用する。本実施形態では、各ファイルの内容をリトリーブするために、ＵＮＩＸ「連結」コマンドであるcatをサービス・ホスト上で使用する。別の好ましい実施形態では、別のオペレーティング・システム内のファイルの内容をリトリーブするための

類似したコマンドを使用し得る。登録する各ファイルの内容をサービス・ホストからリトリーブした後、アドミニストレーション・サーバ上のマネジメント・コンソールにより、各ファイルの内容を周知のスタンダード・パージング技術を使用してパージングする。本実施形態では、コンポーネント・コンフィギュレーション・ファイルはフラットＡＳＣＩＩファイルである。ファイルの内容をパージングすることにより、ファイルのユーザフレンドリーな名前、コンポーネント識別子及び他のコマンド・エクセキューション情報が各ファイルについて識別される。本実施形態では、図４に示すように、コンポーネント・コンフィギュレーション・ディレクトリ内へ保存された３種類のマネジメント・モジュールのそれぞれに関する情報を、この情報は反映している。

【００３４】ステップ５０８では、コンポーネント・コンフィギュレーション・ファイルからパージングしたデータを永続ストレージ領域へ格納する。前記のように、コンポーネント・コンフィギュレーション・ファイルは対応するサービスを開始するために必要な全ての情報を含む。この情報はマネジメント・コンソール・プログラム及びサービス・ホストがそれぞれアクセスできる永続ストレージ上のデータベース内へ現時点で格納されている。システム・アドミニストレータは信頼できる永続データベース内に格納されているサービスのマネジメント・データの内容を変更することにより、サービスをマネジメント・コンソールを通じて管理できる。本実施形態では、ネットワークがダウンした際またはマネジメント・コンソールがアクティブでない際、永続ストレージ上のデータは残され、かつライトウェイト・ディレクトリ・アクセス・プロトコル（ＬＤＡＰ）を通じてアクセスできる。別の好ましい実施形態では、使用するストレージの種類及びネットワークに基づいて、代わりのアクセス・プロトコルを使用し得る。

【００３５】図６及び図７は本発明の１つの実施形態に基づくブラウザ・ホスト上に表示されたグラフィカル・ユーザ・インターフェースのスクリーン・ショットである。図６は「サービスの登録」ユーザ・インターフェースの初期スクリーン・ショットである。ウィンドウ６０２はテキスト・エントリ・サブウィンドウ６０４を有する。システム・アドミニストレータが管理を望んでいるサービスを有するサービス・ホストの名前がテキスト・エントリ・サブウィンドウ６０４へ入力される。本実施形態では、１つのサービス・ホストを入力するための領域が存在する。別の好ましい実施形態では、アドミニストレータは１つを越す数のサービス・ホストを入力できる。アドミニストレータが登録の削除を望むサービスを有するサービス・ホストの名前を入力できるテキスト・エントリ・サブウィンドウ６０６がさらに示されている。選択した項目を入力した後、ユーザがサービス・ホスト上で管理する権限を付与されているサービスのリストをリトリーブするために、ユーザはボタン６０８をクリックし得る。登録を削除できるサービス・ホスト上のサービスのリストをリトリーブするために、アドミニストレータはボタン６１０を押下し得る。

【００３６】図７は「サービスの登録」ユーザ・インターフェースの別のセグメントを示すスクリーン・ショットである。システム・アドミニストレータが管理する権限を付与されているサービスをシステム・アドミニストレータが選択することを、このグラフィカル・ユーザ・インターフェースは可能にする。ユーザ権限付与及びアクセス・コントロールは以下に詳述する。サービスのリスト６１２はウィンドウ６１４に表示されている。リスト６１２はデータベース内に格納されているユーザに関するデータに由来し、かつ図６のフィールド６０４へ入力されたサービス・ホスト上の利用可能なサービスを含む。システム・アドミニストレータは自分が管理またはアクセスを望むサービスを選択する。本実施形態では、これはサン・ニュース（商標）サービス６１６などのサービス名の左側に位置する星印で示す。サービスを選択した後、ユーザは「選択したサービスの登録」バー６１８をクリックする。本実施形態では、これはマウスまたはトラック・ボールなどのポインティング・デバイスを使用して行われ、かつウィンドウ環境内でインプリメントされる。別の好ましい実施形態では、この情報と、以下に別のスクリーンに関連して詳述する情報とを入力するために、簡単なテキストベースのインターフェースまたは更に複雑な音声認識ベースのインターフェースなどの非グラフィカル・ユーザ・インターフェースを使用できる。

【００３７】前記のように、本発明のマネジメント・コンソール・プログラムは、Ｗｅｂベースの分散ネットワーク内の複数のサービス・ホスト上のサービスを管理するためのセントラル・マネジメント・コンソールを有することによって効果を発揮するユーザ認証及びアクセス・コントロールの「シングル・サインオン」方法を含む。現時点では、Ｗｅｂベースのネットワークでは、複数のサービス・ホスト上で利用可能なサービスを維持する責務を有するシステム・アドミニストレータは、アドミニストレータ・クレデンシャルをアドミニストレータがログオンする各サービス・ホストに対して再認証し、かつ伝達する必要がある。ブラウザから操作を行うアドミニストレータは認証のために使用できる単一のユニバーサル・アイデンティティを有していないため、これは事実である。ここで、認証とは、ユーザが特定のサービス・ホストの管理を許される前、より詳細には、特定のサービス・ホスト上のサービスを管理するオペレーションを実施する前に、ユーザのクレデンシャル及び権限付与をベリファイすることを指す。誰がユーザであるのかと、ユーザがサービス・ホスト上で行うことを許可されていることが何であるのかとに関する一致した理解をネ

ットワーク全体に持たせる必要がある。

【００３８】ブラウザ・ホストから実施するサービス・ホスト上のサービスの管理に関連した認証の集中管理及びユーザ・シングル・サインオンを、本発明は可能にする。図２のマネジメント・コンソール・プログラム２１０は、権限付与及びアクセス・コントロール・コンポーネント、即ち、レイヤ２２２を有する。この権限付与レイヤはベリフィケーションのためにデータベース２１２のユーザ・データへアクセスし、かつこの情報をサービス・ホスト２０６上の対応する権限付与レイヤ、即ち、認証レイヤ２３０へ伝達する。システム・アドミニストレータが管理を望んでいる各サービス・ホスト上でのシステム・アドミニストレータの再認証を実施することなく、この情報は処理され、かつそのサービス・ホストへ伝達される。

【００３９】各ユーザに関する情報はデータベース２１２へ格納され、ユーザが入力した情報はこの格納されている情報に対して認証される。この情報、即ち、クレデンシャルがベリファイされた場合、この情報はＣＧＩプログラムを通じてユーザが指し示すサービス・ホストへ伝達される。サービス・ホストがこの情報を受信した後、この情報はシステム・アドミニストレータに代わってデータベース内のユーザ・プロフィールに対して再認証される。換言するならば、ユーザからの介入または特別なステップを伴うことなく、これは「シーンの裏側で」行われる。ユーザはブラウザを通じてマネジメント・コンソールへ一度ログオン（即ち、名前及びパスワードなどの特定の情報を入力）する必要があるのみで、この情報はサービス・ホストへ自動的に伝達される。

【００４０】図８は本発明の１つの実施形態に基づくマネジメント・コンソール・プログラムのユーザのアクセス・コントロール及び認証に関するグラフィカル・ユーザ・インターフェースのスクリーン・ショットである。ウィンドウ７０２は「アドミニストレータの管理」と称する表題を有する。新たなアドミニストレータ及び同アドミニストレータに関連するパスワードと、新たなアドミニストレータが管理を許されるサービスとを入力するために、このウィンドウは使用される。アドミニストレータ名を入力するためのサブウィンドウ７０４と、パスワードをそれぞれ入力及び再入力するためのウィンドウ７０６及びウィンドウ７０８とがウィンドウ７０２内に設けられている。ウィンドウ７０２の下側部分に位置する別のサブウィンドウ７１０は、サブウィンドウ７０４へ入力したアドミニストレータが管理を許されるサービスのリストを含む。マネージング・アドミニストレータ、即ち、「スーパー」アドミニストレータがサービスを選択した後、ボタン７１２が押下される。

【００４１】図９及び図１０は本発明の１つの実施形態に基づくマネジメント・コントロール・プログラム内のアクセス・コントロール及び権限付与を実施するためのプロセスのフローチャートである。実施プロセスはユーザがブラウザ・ホスト（即ち、図２のアドミニストレーション・コンソール２１６）をマネジメント・コンソール・ホストのＵＲＬへポインティングすることから始まる。従って、ステップ８０２では、ユーザはコンソール・ホストのＵＲＬをブラウザ・ホストから入力する。マネジメント・コンソールのＵＲＬはＷｅｂベースのネットワーク内におけるスタンダードＵＲＬの形態をなす。別の好ましい実施形態では、別の種類のロケータをネットワークの種類に基づいて使用できる。

【００４２】ステップ８０４では、アドミニストレータ／ユーザはコンソール・ホスト上のマネジメント・コンソール・プログラムへアクセスするためのユーザ名及びパスワードを要求される。ステップ８０６では、マネジメント・コンソールはステップ８０４で入力されたユーザ名及びパスワードを受理し、ユーザを認証する。このステップは図１１で更に詳細に示す。ユーザが管理する権限を付与された図６の領域６１２に示す選択されたサービス・ホスト上のサービスを、マネジメント・コンソールはデータベース２１２内のデータを調べることによって表示する。これは図２に示すＣＧＩのマネジメント・コンソール・セグメントを使用することによって行われる。本実施形態では、アドミニストレータの権限付与は、アドミニストレータが管理を許されているサービスに関して定義されている。このステップ中、マネジメント・コンソールはアドミニストレータが管理を許されている各サービス及びホストのＵＲＬを作成する。このプロセスは図１１に基づいて以下に詳述する。アドミニストレータが管理できる各サービス・ホスト及びサービスの位置をコンソール・ホストが突き止めることを、ＵＲＬは可能にする。

【００４３】ステップ８０８では、ユーザが管理を望んでいるサービス（即ち、サービス・ホストからの特定のサービス）のインスタンスをユーザは選択する。サービスは幾つかの異なるサービス・ホスト上に存在し得る。従って、ユーザはサービスのインスタンスを特定のサービス・ホストから選択する必要がある。ユーザ・フレンドリーな名前を選択することにより、ユーザはステップ８０６で作られた複数のＵＲＬのうちの選択された１つを有する。ステップ８１０では、マネジメント・コンソール・ホストはＣＧＩのサーブレットＣＧＩコンポーネントを開始する。本実施形態では、これはユーザ・クレデンシャルまたはプロフィールをデータベース内のユーザの認証及びアクセス・コントロール・データと比較することによって行われる。マネジメント・コンソール・ホスト２０８を通すことなくサービス・ホスト上のサービスの管理を試みるユーザに対する特別な予防措置として、このベリフィケーションはサーブレットＣＧＩ２２４がサービス・ホストへのコネクションを確立する前に実施される。これはネットワーク環境であるため、ユー

ザはコンソール・ホスト・ベリフィケーション・ステップをバイパスし、かつ図2のブラウザ・ホスト216からサービス・ホスト上のサービスへアクセスする代わりに、クライアント・マシンからサービス・ホスト上のサービスへ直接アクセスを試み得る。このため、ユーザ・クレデンシャルをサーブレットCGIを通じてデータベース212内に格納されているユーザ・データと比較する。

【0044】ステップ812では、ユーザ・クレデンシャルをユーザが指し示すサービス・ホストへ伝達するために、サーブレットCGIは標準の手順を使用する。本実施形態では、サービス・ホストがデータを受信した後、サービス・ホストはこのデータをデータベース内のデータと比較することにより認証及びアクセス・コントロールを実施する。別の好ましい実施形態では、マネジメント・コンソール・プログラムをインプリメントする特定のネットワーク上で使用可能な独立したセキュリティ機能に基づいて、このステップは不要となることがある。この再認証はユーザからの介入を受けることなく行われるとともに、ユーザがサービス・ホストへ直接ログオンし、これによって、マネジメント・コンソール・ホストの認証及びアクセス・コントロール・レイヤの迂回を試みないことを保証するために実施される。従って、特別な操作の実施をユーザへ要求することなくデータベースに対する第2のチェックを実施することにより、マネジメント・コンソールはネットワーク内のサービスの安全な管理を保証できる。再認証がステップ814で成功した場合、ステップ816に示すように、コンソール・ホスト上のマネジメント・コンソール・プログラムは、ユーザが選択されたサービスに対するマネジメント・オペレーションをブラウザから実施することを可能にする。この時点で、実施プロセスは完了する。再認証が成功しなかった場合、ユーザは選択されたサービスを管理する権限を付与されず、かつログイン・スクリーンを再び見ることになる。

【0045】図11は図9のステップ806の詳細を示すフローチャートである。ステップ806では、ユーザを認証し、ユーザがアクセスする権限を付与されたサービスを決定し、これらのサービスのURLをそれぞれ作成する。ステップ902では、マネジメント・コンソール・ホストはユーザに関する情報をデータベースからリトリーブすることによりユーザを認証する。この情報はユーザ名及びパスワードからなる。ユーザ名及びパスワードをベリファイした後、ユーザが管理する権限を付与されたサービスのリストを導出する。ステップ904では、コンソール・ホストはステップ902でベリファイされたユーザ・クレデンシャルを用いてCGIプログラムのマネジメント・コンソール・セグメント226を開始する。前記のように、これはサービス・ホストに対するリンクを確立する最初のステップである。

【0046】CGIの別のコンポーネントはサービス・ホストに対するコネクションを確立するために使用するサーブレットCGI（図2のアイテム224）である。ステップ906では、ユーザが管理する権限を付与されているサービスのリストを獲得するために、マネジメント・コンソールCGIは図2のデータベース212を必要とする。これらのサービスへのリンクはリスト上の全てのサービスへのURLの形態で作成される。ユーザ名と、パスワードと、レベル（例：スーパー・システム・アドミニストレータ）と、ユーザが管理を許されているサービスのリストとを含む情報を有する各ユーザのエントリをデータベースは含む。スーパー・システム・アドミニストレータは全てのサービスを管理し、かつ他のユーザ（例：ジュニア・システム・アドミニストレータ）のアクセス・コントロール・パラメータを定義し得る。サービスのリストはサービスのURLの代わりにサービスの「ユーザフレンドリー」な名前を含む（この名前はデータベースにも含まれる）。次いで、コントロールは図9のステップ806へ戻り、ユーザは自分が管理を望むサービスをサービスのリストから選択する。

【0047】本発明はコンピュータ・システム内に格納されたデータを使用する様々なコンピュータによって実現される動作を使用する。これらの動作は物理量の物理操作を必要とする動作を含む（但し、同動作に限定されない）。一般的に、必ずしも必要でないが、これらの量は格納、転送、結合、比較及び他の操作が可能な電気信号または磁気信号の形態をなす。本発明の一部を構成する本明細書中に開示する動作は、効果的な機械動作である。実行される操作は生成（producing）、識別（identifying）、実行（running）、決定（determining）、比較（comparing）、実行（executing）、ダウンローディング（downloading）または検出（detecting）等の用語で示されることが多い。特に、共通の用法を確立するために、これらの電気信号または磁気信号をビット、値、エレメント、変数、文字またはデータ等として示すと都合が良い。しかし、これらの用語またはこれらに類似する用語の全ては適切な物理量に付随すべきであり、かつこれらの物理量に適用された都合の良いラベルにすぎない点を覚えておく必要がある。

【0048】更に、本発明は前記の動作を実行するためのブラウザ・ホスト216及びマネジメント・コンソール・ホスト208などのデバイス、システムまたは装置に関する。システムは要求された目的のために特別に構築可能である。また、システムは汎用コンピュータとすることが可能である。汎用コンピュータに格納されたコンピュータ・プログラムによって、同汎用コンピュータは選択的に作動または設定される。前記の複数のプロセスは特定のコンピュータまたは他のコンピューティング装置に固有のものではない。特に、本明細書の開示内容に基づいて記述されたプログラムを様々な汎用コンピュ

ータと併用可能である。これに代えて、要求された動作を実行するために、更に特別なコンピュータ・システムを形成することは更に都合が良い。

【００４９】図１２は本発明の１つの実施形態に基づく処理の実施に適した汎用コンピュータ・システム１０００のブロック図である。認証及びアクセス・コントロール・レイヤを含むマネジメント・コンソール・プログラムは、この種の汎用コンピュータ上に存在し得る。更に、ブラウザ・ホスト２１６は、この種の汎用コンピュータであり得る。図１２は汎用コンピュータ・システムの１つの例を示す。本発明の処理を実施するために、他のコンピュータ・システム・アーキテクチャ及びコンフィギュレーションを使用し得る。以下に詳述する様々なサブシステムからなるコンピュータ・システム１０００は少なくとも１つのマイクロプロセッサ・サブシステム（中央処理装置、即ち、ＣＰＵとも称される）１００２を含む。即ち、ＣＰＵ１００２はシングルチップ・プロセッサまたは複数のプロセッサによって実現し得る。ＣＰＵ１００２はコンピュータ・システム１０００のオペレーションを制御する汎用デジタル・プロセッサである。メモリからリトリーブした命令を使用して、ＣＰＵ１００２は入力データの受信及び操作と、出力デバイス上でのデータの出力及び表示とを制御する。

【００５０】ＣＰＵ１００２は一般的にランダム・アクセス・メモリ（ＲＡＭ）からなる第１の一次ストレージ１００４へメモリ・バス１００８を通じて両方向接続されている。更に、ＣＰＵ１００２は一般的にリード・オンリ・メモリ（ＲＯＭ）からなる第２の一次ストレージ領域１００６へメモリ・バス１００８を通じて単方向接続されている。当該技術分野でよく知られているように、一次ストレージ１００４は汎用ストレージ領域及び作業メモリとして使用可能であり、さらには入力データ及び処理済みデータを格納するためにも使用できる。更に、ＣＰＵ１００２上の動作を処理するためのデータ及び命令を格納する以外に、一次ストレージ１００４はプログラミング命令及びデータを例えばデータベース２１２などの階層データベースの形態で格納可能である。更に、データ及び命令をメモリ・バス１００８を通じて両方向で高速転送するために、一次ストレージ１００４は一般的に使用される。同様に、当該技術分野でよく知れられているように、ＣＰＵ１００２がその機能を果たすために使用する基本動作命令、プログラム・コード、データ及びオブジェクトを一次ストレージ１００６は一般的に含む。データ・アクセスが両方向または単方向のいずれを必要とするかなどの条件に基づいて、一次ストレージ・デバイス１００４，１００６は以下に詳述する任意の適切なコンピュータ読み取り可能ストレージ媒体を含み得る。ＣＰＵ１００２は頻繁に必要となるデータをキャッシュ・メモリ１０１０内へ超高速で直接リトリーブし、かつ格納できる。

【００５１】取り外し可能大容量ストレージ・デバイス１０１２はコンピュータ・システム１０００のための別のデータ・ストレージ能力を提供し、かつペリフェラル・バス１０１４を通じてＣＰＵ１００２へ両方向または単方向で接続されている。例えば、ＣＤ－ＲＯＭとして一般的に知られている特定の取り外し可能大容量ストレージ・デバイスはデータを単方向でＣＰＵ１００２へ送信する。その一方、フロッピー・ディスクはデータを両方向でＣＰＵ１００２へ送信し得る。ストレージ１０１２は磁気テープ、フラッシュ・メモリ、搬送波に組み込まれた信号、ＰＣカード、ポータブル大容量ストレージ・デバイス、ホログラフィック・ストレージ・デバイス及び他のストレージ・デバイス等のコンピュータ読み取り可能媒体を更に含み得る。固定大容量ストレージ１０１６は別のデータ・ストレージ能力を提供し、かつペリフェラル・バス１０１４を通じてＣＰＵ１００２へ両方向で接続されている。大容量ストレージ１０１６の最も一般的な例としては、ハード・ディスク・ドライブが挙げられる。一般的に、これらの媒体へのアクセスは一次ストレージ１００４，１００６へのアクセスより遅い。ＣＰＵ１００２が頻繁に使用しない他のプログラミング命令及びデータ等を大容量ストレージ１０１２，１０１６は一般的に格納する。必要に応じて、大容量ストレージ１０１２，１０１６内に保持された情報は、一次ストレージ１００４（例：ＲＡＭ）の一部を構成するバーチャル・メモリとして標準的に組込み可能である。

【００５２】ストレージ・サブシステムへのＣＰＵ１００２のアクセスを提供する以外に、ペリフェラル・バス１０１４は他のサブシステム及びデバイスへのアクセスを提供するために使用できる。本実施形態では、これらは、ディスプレイ・モニタ１０１８及びアダプタ１０２０、プリンタ・デバイス１０２２、ネットワーク・インターフェース１０２４、補助入出力装置インターフェース１０２６、サウンド・カード１０２８及びスピーカー１０３０、並びに必要とされる他のサブシステムを含む。

【００５３】図示するように、ネットワーク接続を使用することにより、ネットワーク・インターフェース１０２４はＣＰＵ１００２を別のコンピュータ、コンピュータ・ネットワークまたはテレコミュニケーション・ネットワークへ接続可能にする。前記の方法のステップを実行するうえで、ＣＰＵ１００２はデータ・オブジェクトまたはプログラム命令などの情報を別のネットワークからネットワーク・インターフェース１０２４を通じて受信するか、または情報をネットワーク・インターフェース１０２４を通じて別のネットワークへ送信し得る。ＣＰＵで実行する複数の命令のシーケンスに代表される情報は、搬送波に組み込まれたコンピュータ・データ信号などの形態で別のネットワークに対して送受信可能である。インターフェース・カードまたはこれに類似するデ

バイスと、ＣＰＵ１００２によって実行される適切なソフトウェアとは、コンピュータ・システム１０００を外部ネットワークへ接続し、かつデータを標準プロトコルに基づいて転送するために使用できる。即ち、本発明の方法はＣＰＵ１００２上で単独で実行し得る。その一方、処理の一部を共有する遠隔ＣＰＵと協動することにより、本発明の方法をインターネット、イントラネットワーク若しくはローカル・エリア・ネットワーク等のネットワークを通じて実行し得る。別の大容量ストレージ・デバイス（図示略）をネットワーク・インターフェース１０２４を通じてＣＰＵ１００２へ接続し得る。

【００５４】マイクロホン、タッチ・ディスプレイ、トランスデューサ・カード・リーダー、テープ・リーダー、音声認識装置、手書き文字認識装置、バイオメトリクス・リーダー、カメラ、ポータブル大容量ストレージ・デバイス及び他のコンピュータ等の装置に対するＣＰＵ１００２によるデータの送受信を可能にする汎用インターフェース及びカスタム・インターフェースを補助入出力装置インターフェース１０２６は表す。

【００５５】キーボード１０３６またはポインタ・デバイス１０３８からの入力を受信し、さらにはデコードしたシンボルをキーボード１０３６またはポインタ・デバイス１０３８からＣＰＵ１００２へ送信するために、キーボード・コントローラ１０３２がローカル・バス１０３４を通じてＣＰＵ１００２へ接続されている。ポインタ・デバイスはマウス、スタイラス、トラック・ボールまたはタブレットであり得る。そして、ポインタ・デバイスはグラフィカル・ユーザ・インターフェースとのインターフェースに効果的である。

【００５６】更に、本発明の実施形態は様々なコンピュータ実現動作を実施するためのプログラム・コードを含むコンピュータ読み取り可能媒体を有するコンピュータ・ストレージ・プロダクトに関する。コンピュータ読み取り可能媒体は、コンピュータ・システムによる後からの読み取りが可能なデータを格納し得る任意のデータ・ストレージ・デバイスである。媒体及びプログラム・コードは本発明の目的のために特別に設計され、かつ構築されたものであるか、またはコンピュータ・ソフトウェア技術分野の当業者によく知られたものであり得る。コンピュータ読み取り可能媒体の例としては、ハード・ディスク、フロッピー・ディスク及び磁気テープなどの磁気媒体と、ＣＤ－ＲＯＭディスクなどの光媒体と、光フロッピー・ディスクなどの磁気光媒体と、特定用途向け集積回路（ＡＳＩＣ）、プログラム可能論理回路（ＰＬＤ）、ＲＯＭデバイス及びＲＡＭデバイスなどの特別に構成されたハードウェア・デバイスとを含めた前記の全ての媒体が挙げられる（但し、これらに限定されない）。コンピュータ読み取り可能媒体は搬送波に組み込まれたデータ信号として結合コンピュータ・システムのネットワーク上に分散させ得る。従って、コンピュータ読み取り可能コードは分散した形態で格納及び実行できる。プログラム・コードの例としては、コンパイラなどによって形成されたマシン・コード、またはインタプリタを使用して実行できる高レベル・コードを含むファイルが挙げられる。

【００５７】前記のハードウェア・エレメント及びソフトウェア・エレメントが標準的なデザイン及び構成を有することを当業者は認める。本発明に適した他のコンピュータ・システムは別のサブシステムまたは更に少ない数のサブシステムを含み得る。更に、メモリ・バス１００８、ペリフェラル・バス１０１４及びローカル・バス１０３４は複数のサブシステムをリンクするために使用する任意の相互接続方式の実例である。例えば、ローカル・バスはＣＰＵを固定大容量ストレージ１０１６及びディスプレイ・アダプタ１０２０へ接続するために使用可能である。図１２に示すコンピュータ・システムは本発明に適したコンピュータ・システムの例である。サブシステムの別のコンフィギュレーションを有する他のコンピュータ・アーキテクチャを使用し得る。

【００５８】以上、理解を容易にする目的で、本発明をある程度詳しく説明したが、特定の変更及び修正を本発明の請求の範囲内で実施しても良い。更に、本発明の方法及び装置の両方を実現する他の方法があることを認識する必要がある。例えば、Ｗｅｂサーバをアドミニストレーション・サーバとして使用することによって、本発明を説明したが、マネジメント・コンソール・プログラムを実行するために、Ｗｅｂベースでないサーバを使用し得る。別の例では、データベース２１２を単一の永続データベースとせずに、コンソール・ホスト及び複数の異なるサービス・ホストへ格納された分散データベースとし得る。更に別の例では、データをデータベース２１２または永続ストレージ領域に格納されたフラット・ファイルからリトリーブするために、ＬＤＡＰ以外のデータ・リトリーバル・プロトコルを使用できる。別の例では、ディスカバー・ルーチンをコンソール・ホストで後から実行する代わりに、サービスがインストールされている状態で、ディスカバー・ルーチンをサービス・ホスト上でローカルに実行しても良い。従って、本明細書に開示した複数の実施形態は例示を目的とするものであって、限定を目的とするものではない。更に、本発明は本明細書に開示する詳細部分に限定されることなく、請求の範囲及びそれに等価な範囲内で変更し得る。

【図面の簡単な説明】

【図１】エンドユーザがアクセスできる複数のサービスを有するコンピュータ・ネットワークであって、本発明の自動管理能力を持たないアドミニストレーション・サーバへ接続されたコンピュータ・ネットワークのブロック図である。

【図２】本発明の１つの実施形態に基づくコンピュータ・ネットワークのサーバ側コンポーネントのブロック図

である。

【図３】新たなサービスをネットワーク上へ登録する本発明の１つの実施形態に基づく方法の概要を示すフローチャートである。

【図４】本発明の１つの実施形態に基づくサービスを登録する図３のステップ３０４の詳細を示すフローチャートである。

【図５】本発明の１つの実施形態に基づく図３のステップ３０６の詳細を示すフローチャートである。

【図６】本発明の１つの実施形態に基づくブラウザ・ホスト上に表示されたグラフィカル・ユーザ・インターフェースのスクリーン・ショットである。

【図７】本発明の１つの実施形態に基づくブラウザ・ホスト上に表示されたグラフィカル・ユーザ・インターフェースのスクリーン・ショットである。

【図８】本発明の１つの実施形態に基づくマネジメント・コンソール・プログラムのユーザのアクセス・コントロール及び認証に関するグラフィカル・ユーザ・インターフェースのスクリーン・ショットである。

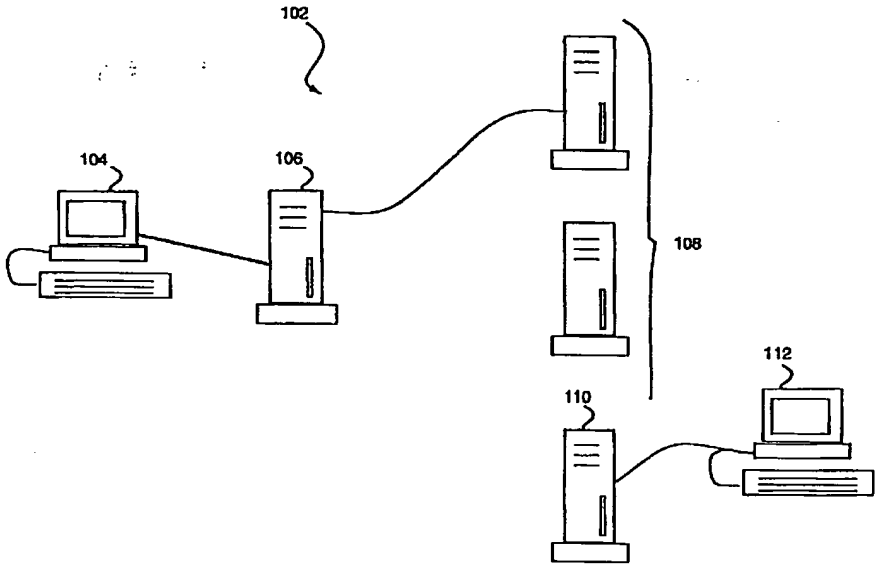
【図９】本発明の１つの実施形態に基づくマネジメント・コントロール・プログラムにおけるアクセス・コントロール及び権限付与を実施するためのプロセスのフローチャートである。

【図１０】本発明の１つの実施形態に基づくマネジメント・コントロール・プログラムにおけるアクセス・コントロール及び権限付与を実施する方法のプロセスのフローチャートである。

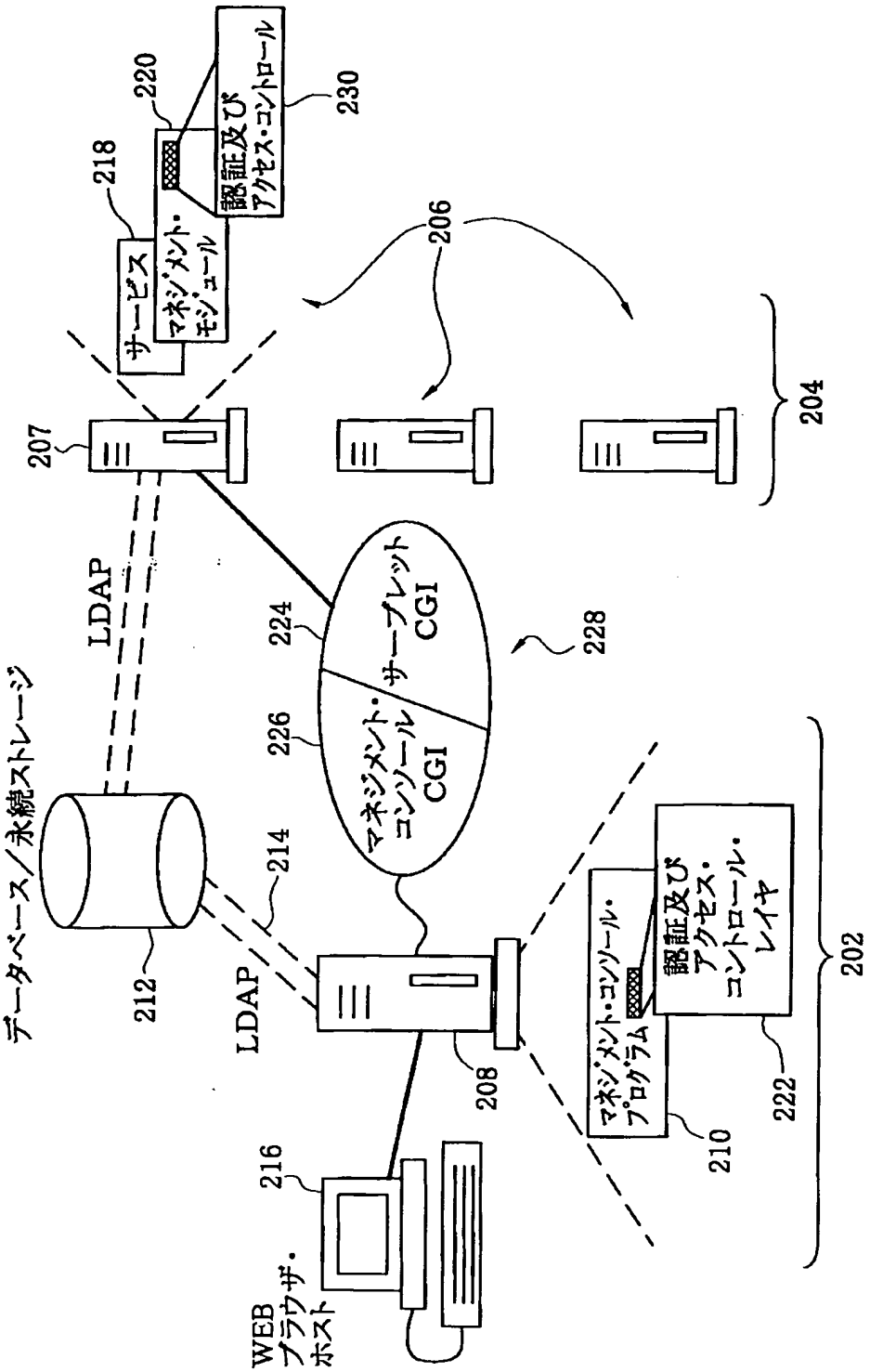
【図１１】図９のステップ８０６の詳細を示すフローチャートである。

【図１２】本発明の実施形態を実現することに適した一般的なコンピュータ・システムのブロック図である。

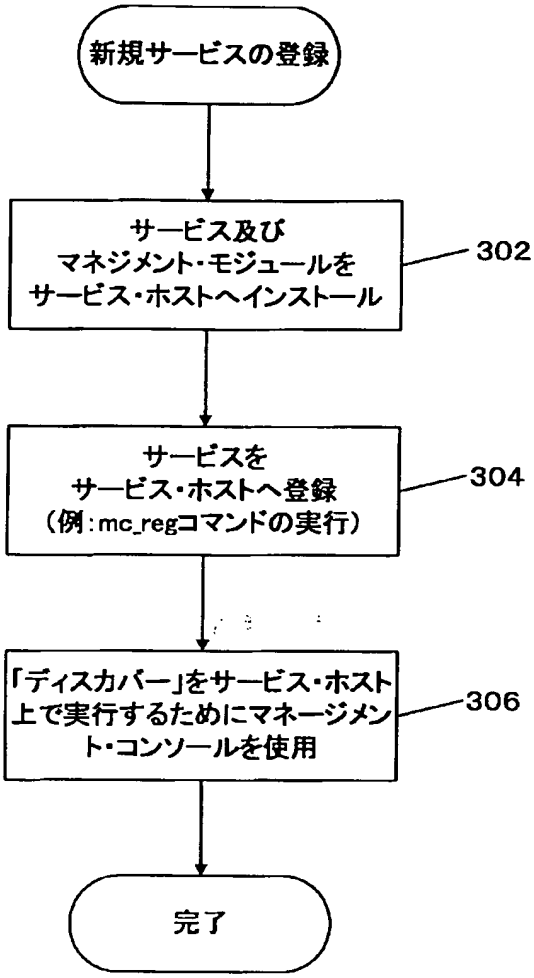
【図１】



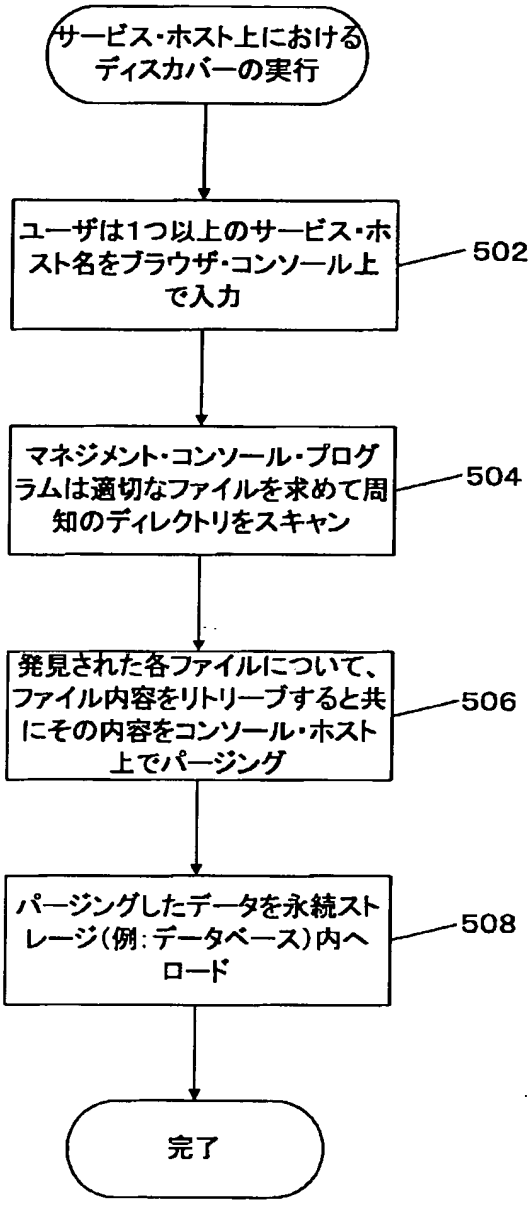
【図2】


データベース／永続ストレージ
212
LDAP
214
LDAP
216
WEB
ブラウザ・
ホスト
208
マネジメント・
コンソール
CGI
226
サーブレット
CGI
224
228
マネジメント・コンソール・
プログラム
210
認証及び
アクセス・
コントロール・
レイヤ
222
202
207
サービス
218
マネジメント・
モジュール
220
認証及び
アクセス・コントロール
230
206
204

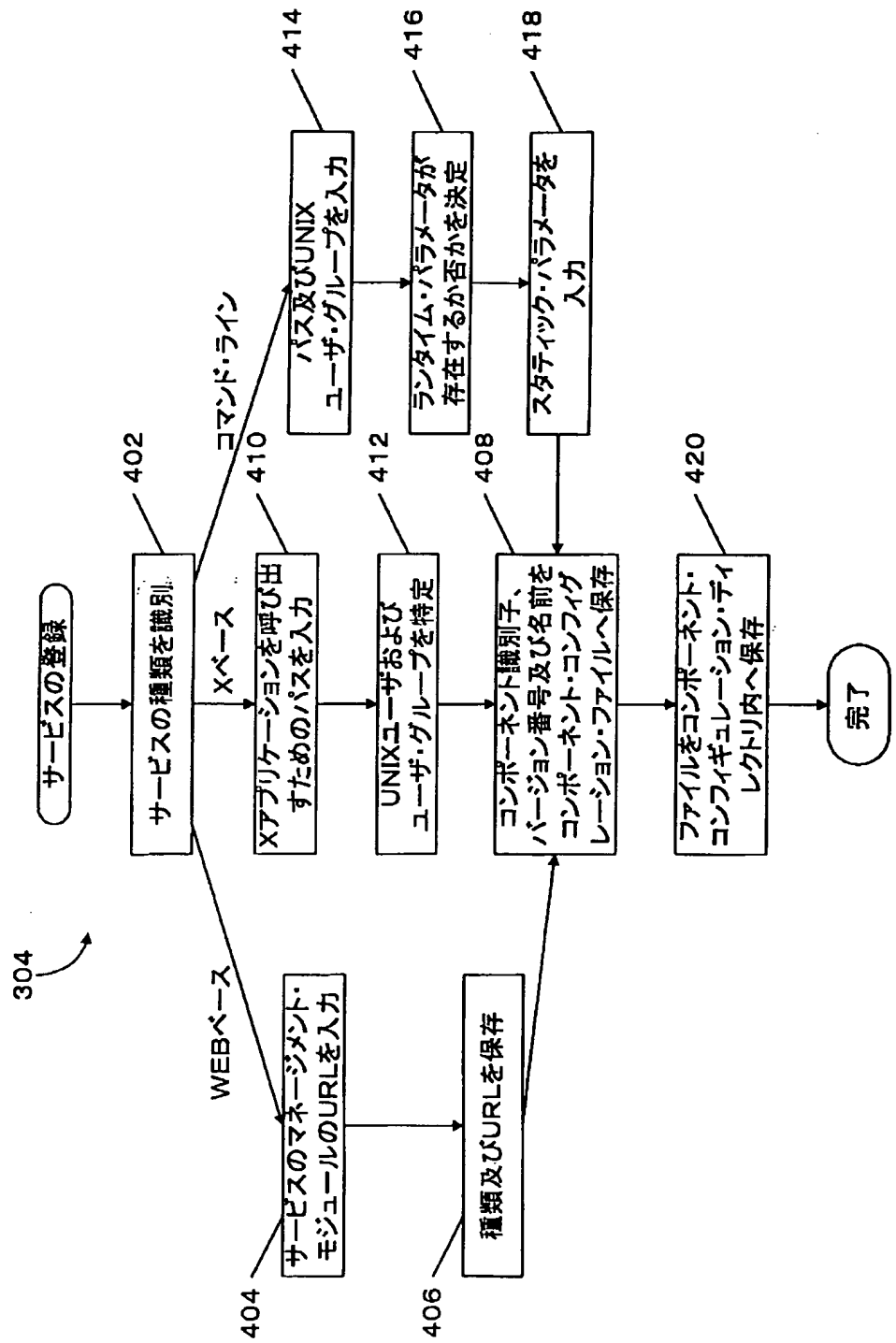
【図3】


新規サービスの登録
サービス及び
マネジメント・モジュールを
サービス・ホストへインストール
302
サービスを
サービス・ホストへ登録
(例:mc_regコマンドの実行)
304
「ディスカバー」をサービス・ホスト
上で実行するためにマネージメン
ト・コンソールを使用
306
完了

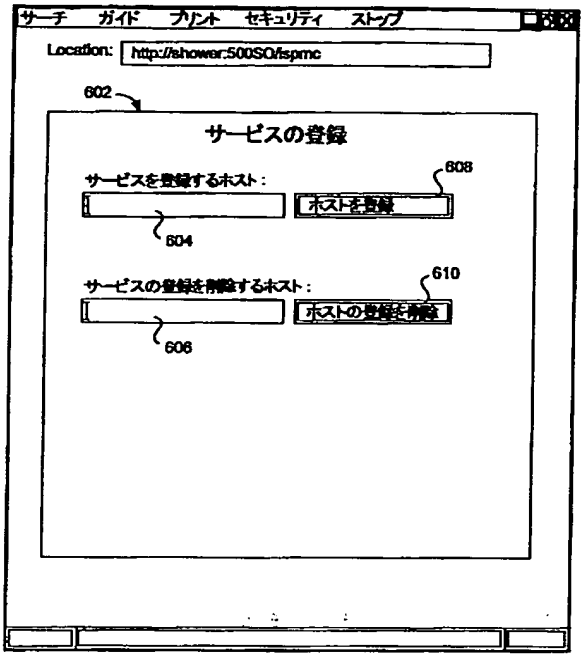
【図5】


サービス・ホスト上における
ディスカバーの実行
ユーザは1つ以上のサービス・ホ
スト名をブラウザ・コンソール上
で入力
502
マネジメント・コンソール・プログ
ラムは適切なファイルを求めて周
知のディレクトリをスキャン
504
発見された各ファイルについて、
ファイル内容をリトリーブすると共
にその内容をコンソール・ホスト
上でパージング
506
パージングしたデータを永続スト
レージ(例:データベース)内へ
ロード
508
完了

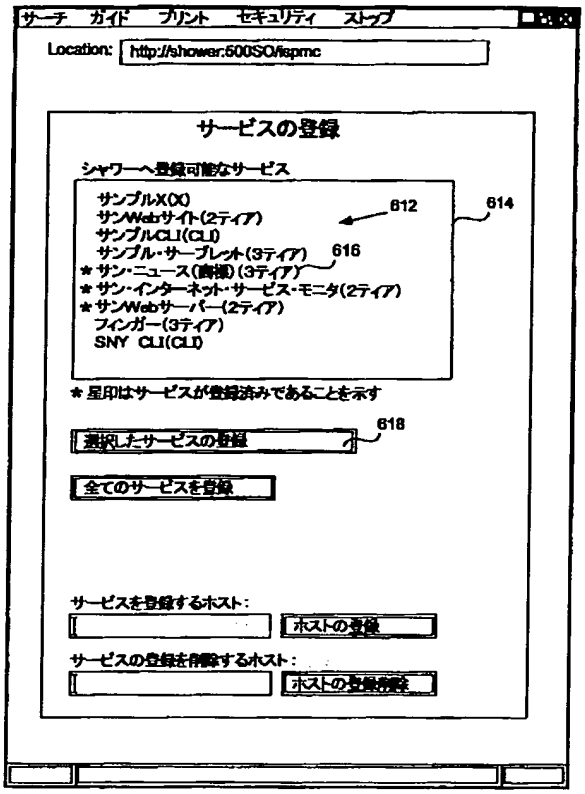
【図4】


サービスの登録
サービスの種類を識別
402
コマンド・ライン
Xベース
WEBベース
304
Xアプリケーションを呼び出すためのパスを入力
410
パス及びUNIXユーザ・グループを入力
414
UNIXユーザおよびユーザ・グループを特定
412
ランタイム・パラメータが存在するか否かを決定
416
スタティック・パラメータを入力
418
コンポーネント識別子、バージョン番号及び名前をコンポーネント・コンフィグレーション・ファイルへ保存
408
ファイルをコンポーネント・コンフィギュレーション・ディレクトリ内へ保存
420
完了
サービスのマネージメント・モジュールのURLを入力
404
種類及びURLを保存
406

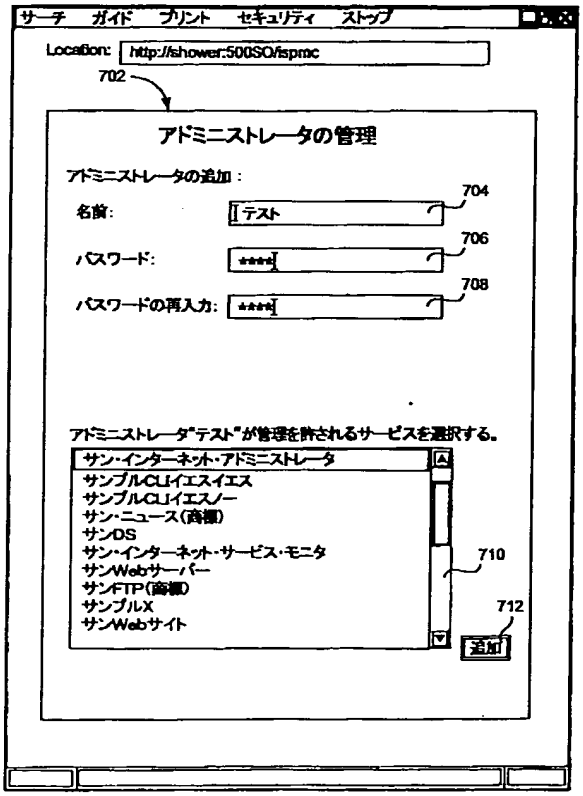
【図6】


サーチ ガイド プリント セキュリティ ストップ
Location: http://shower:500SO/ispmc
602
サービスの登録
サービスを登録するホスト:
608
ホストを登録
604
サービスの登録を削除するホスト:
610
ホストの登録を削除
606

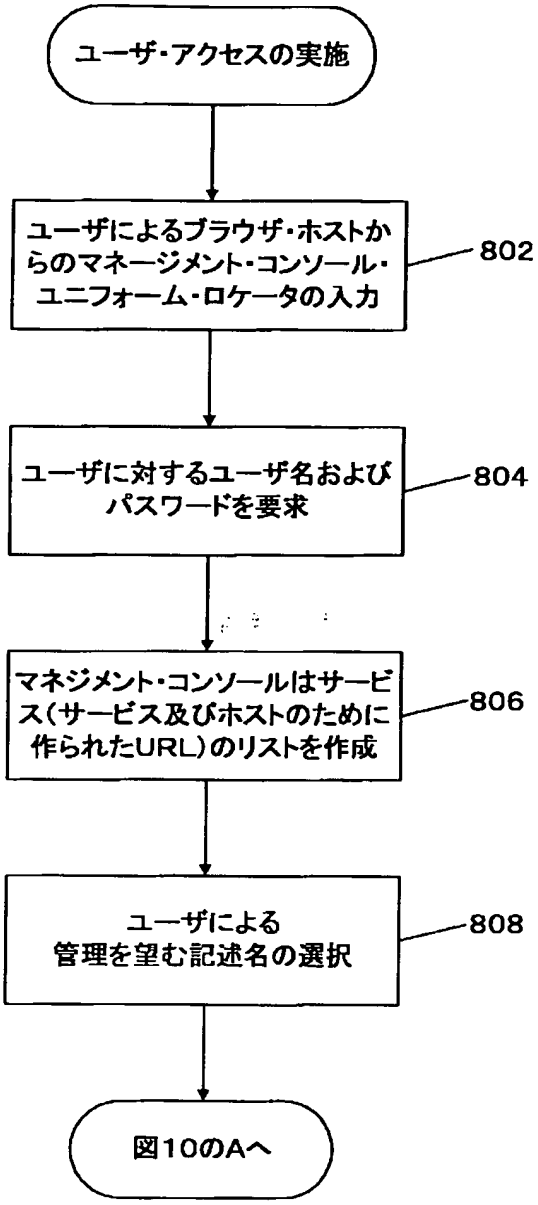
【図7】


サーチ ガイド プリント セキュリティ ストップ
Location: http://shower:500SO/ispmc
サービスの登録
シャワーへ登録可能なサービス
サンプルX(X)
サンWebサイト(2ティア)
サンプルCLI(CLI)
サンプル・サーブレット(3ティア)
* サン・ニュース(商標)(3ティア)
* サン・インターネット・サービス・モニタ(2ティア)
* サンWebサーバー(2ティア)
フィンガー(3ティア)
SNY CLI(CLI)
612
614
616
* 星印はサービスが登録済みであることを示す
618
選択したサービスの登録
全てのサービスを登録
サービスを登録するホスト:
ホストの登録
サービスの登録を削除するホスト:
ホストの登録削除

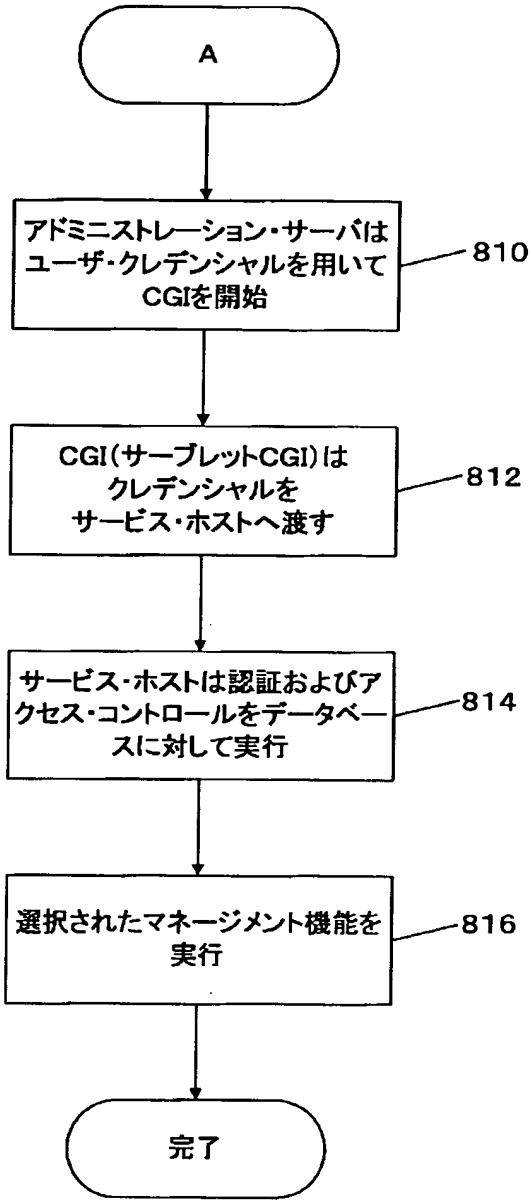
【図8】


サーチ ガイド プリント セキュリティ ストップ
Location: http://shower:500SO/ispmc
702
アドミニストレータの管理
アドミニストレータの追加:
名前:
テスト
704
パスワード:
****
706
パスワードの再入力:
****
708
アドミニストレータ"テスト"が管理を許されるサービスを選択する。
サン・インターネット・アドミニストレータ
サンプルCLIイエスイエス
サンプルCLIイエスノー
サン・ニュース(商標)
サンDS
サン・インターネット・サービス・モニタ
サンWebサーバー
サンFTP(商標)
サンプルX
サンWebサイト
710
712
追加

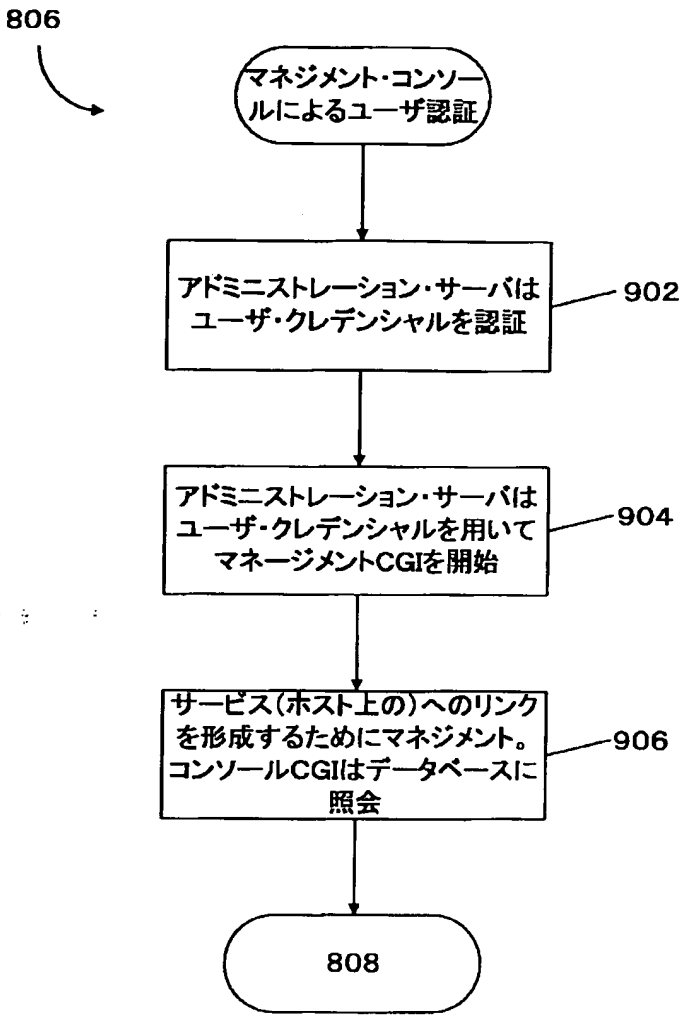
【図9】


ユーザ・アクセスの実施
ユーザによるブラウザ・ホストからのマネージメント・コンソール・ユニフォーム・ロケータの入力
802
ユーザに対するユーザ名およびパスワードを要求
804
マネジメント・コンソールはサービス(サービス及びホストのために作られたURL)のリストを作成
806
ユーザによる管理を望む記述名の選択
808
図10のAへ

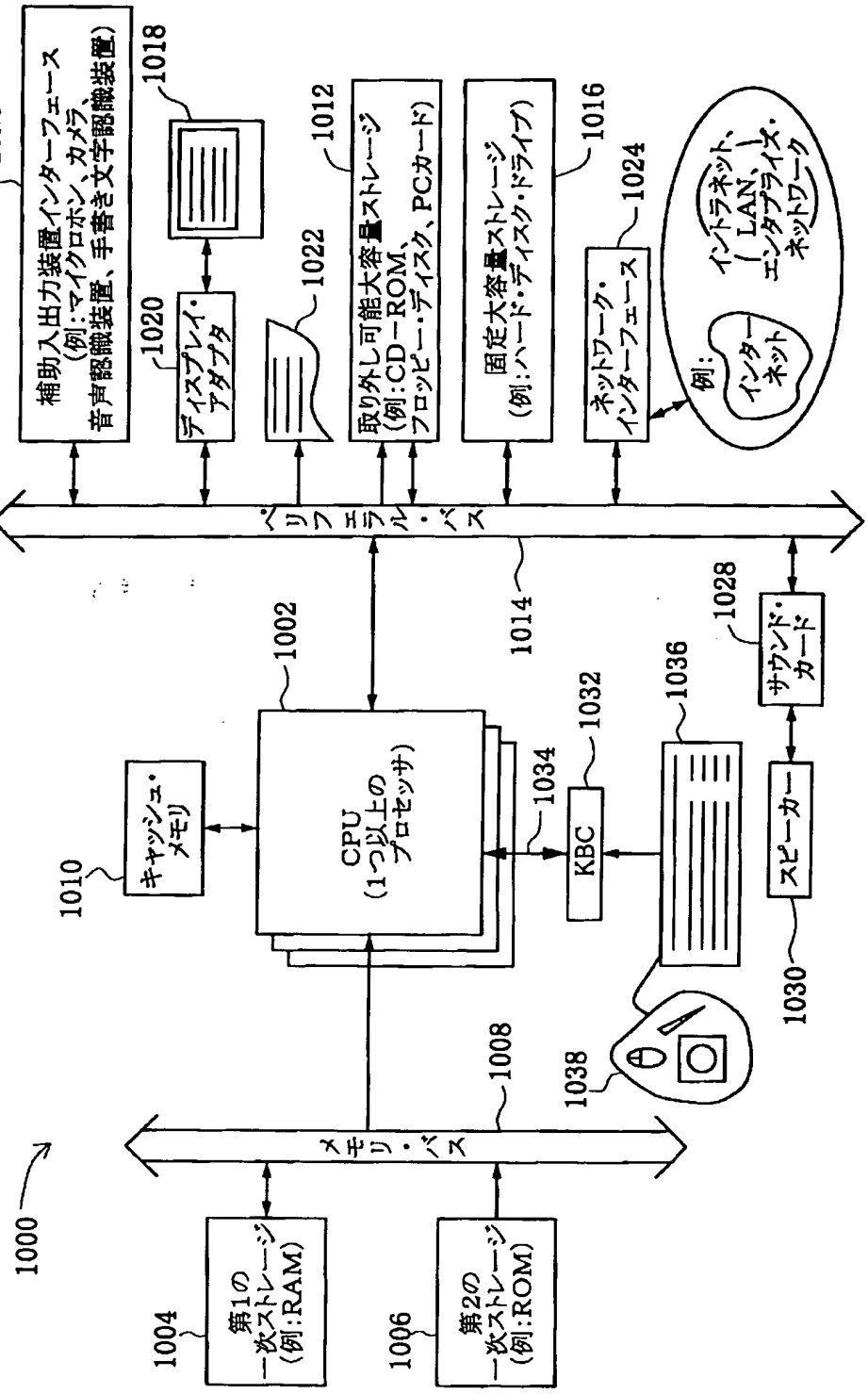
【図10】


A
アドミニストレーション・サーバはユーザ・クレデンシャルを用いてCGIを開始
810
CGI(サーブレットCGI)はクレデンシャルをサービス・ホストへ渡す
812
サービス・ホストは認証およびアクセス・コントロールをデータベースに対して実行
814
選択されたマネージメント機能を実行
816
完了

【図11】


806
マネジメント・コンソールによるユーザ認証
アドミニストレーション・サーバはユーザ・クレデンシャルを認証
902
アドミニストレーション・サーバはユーザ・クレデンシャルを用いてマネージメントCGIを開始
904
サービス(ホスト上の)へのリンクを形成するためにマネジメント。コンソールCGIはデータベースに照会
906
808

【図12】


1000
1002
CPU
(1つ以上の
プロセッサ)
1004
第1の
一次ストレージ
(例:RAM)
1006
第2の
一次ストレージ
(例:ROM)
1008
メモリ・バス
1010
キャッシュ・
メモリ
1012
取り外し可能大容量ストレージ
(例:CD-ROM、
フロッピー・ディスク、PCカード)
1014
ペリフェラル・バス
1016
固定大容量ストレージ
(例:ハード・ディスク・ドライブ)
1018
1020
ディスプレイ・
アダプタ
1022
1024
ネットワーク・
インターフェース
1026
補助入出力装置インターフェース
(例:マイクロホン、カメラ、
音声認識装置、手書き文字認識装置)
1028
サウンド・
カード
1030
スピーカー
1032
KBC
1034
1036
1038
例:
インターネット
イントラネット、
LAN、
エンタプライズ・
ネットワーク

フロントページの続き

(71)出願人 591064003
901 SAN ANTONIO ROAD
PALO ALTO, CA 94303, U. S. A.
(72)発明者 エイプリル・エス. ・チャング
アメリカ合衆国 カリフォルニア州94024
ロス・アルトス, アルフォード・アベニュー, 1872
(72)発明者 アンドリュー・アール. ・ラージ
アメリカ合衆国 カリフォルニア州95076
ラ・セルバ・ビーチ, カミノ・アル・バランコ, 285
(72)発明者 アラン・スナイダー
アメリカ合衆国 カリフォルニア州94306
パロ・アルト, ブライアーウッド・ウェイ, 4160

【外国語明細書】

1 TITLE OF INVENTION

Authentication and Access Control in a Management Console Program for Managing Services in a Computer Network

2 CLAIMS

1. A method of securing access to the administration of a plurality of distinct services residing on one or more service host computers from an administration server computer connected to the one or more service host computers, there being a service manager residing on the administration server computer, the method comprising:

providing a selected user identifier and a corresponding private keyword, the user identifier being arranged to identify a user having administrative access to at least one of the distinct services;

authenticating the user by comparing the selected user identifier and the corresponding private keyword against a plurality of user identifiers and private keywords stored in a persistent storage area, the comparing performed under control of the service manager;

deriving a list of services to which the user associated with the user identifier has administrative access;

when a request is made to administer a selected one of the services in the derived list of services, verifying at the service host computer associated with the selected service that the user associated with the selected user identifier is permitted to access the selected service by examining access control data associated with the selected user identifier in the persistent storage area; and

transferring one or more management files on the service host computer to the administration server thereby facilitating manipulation of the management files utilizing the service manager.

2. A method as recited in claim 1 wherein the administration server comp uter is connected to an administration client computer suitable for runn ing a browser program and wherein the selected user identifier and the c orresponding private keyword are provided over a communications connecti on between the administration client computer and the administration ser ver computer, the communications connections among the administration se rver computer, the administration client computer and the one or more se rvice host computers utilizing an internet protocol.

3. A method as recited in claim 1 wherein providing a selected user iden tifier and a corresponding private keyword further comprises logging on to the service manager through the administration client computer.

4. A method as recited in claim 1 wherein authenticating the user furthe r comprises utilizing a lightweight directory access protocol to communi cate the user identifier and corresponding private keyword to the persis tent storage area.

5. A method as recited in claim 1 wherein each user identifier has a cor responding user profile that represents a global user identity correspon ding to a particular service manager user.

6. A method as recited in claim 1 wherein deriving a list of services fu rther comprises searching the persistent storage area, the persistent st orage area containing a user profile database including, for each user, a user access level, a list of allowable services, and a password.

7. A method as recited in claim 1 wherein verifying at the service host

computer that the user associated with the selected user identifier is p
ermitted to access the selected service from the list of services furthe
r comprises communicating the selected user identifier and the correspon
ding private keyword to the host server computer using a common gateway
interface.

8. A method as recited in claim 1 wherein the service host computer cont
ains an authentication and access control segment.

9. A method as recited in claim 1 wherein the selected user identifier a
nd the corresponding private keyword are automatically passed to the one
or more service host computers for use in .

10. A method as recited in claim 1 further comprising displaying the lis
t of services in a user interface displayed on the administration client
computer.

11. A method as recited in claim 1 further comprising constructing a ser
vice locator by the management console program for locating a service on
a host server computer.

12. A method as recited in claim 1 wherein transferring one or more mana
gement files on the host server to the administration server further com
prises initiating a common gateway interface on the administration serve
r computer thereby enabling the transfer of one or more management files
and a plurality of operating system commands.

13. A system for securing administration of services residing on one or
more service host computers from an administration server computer, the

administration server computer connected to an administration client having a browser-type program and to the one or more service host computers using an Internet protocol, the system comprising:

a user profile data repository for storing data relating to user privileges, the data including, for each user, a user access level, a list of services, and a password;

a service manager subcomponent of a communication interface residing on the administration server computer for accepting a user identifier and a corresponding keyword and passing the user identifier and the corresponding keyword to the user profile data repository;

a component configuration directory suitable for residing on the one or more service hosts containing component configuration files for storing management modules associated with the plurality of services, the management modules containing management data utilized in administering the plurality of services;

a service host subcomponent of the communication interface residing on the administration server computer for accepting the user identifier and the corresponding keyword and passing the user identifier and the corresponding keyword to the plurality of service host computers for verification by examining data relating to user privileges stored in the user profile data repository.

14. A system for securing access to the administration of a plurality of distinct services residing on one or more service host computers from a administration server computer connected to the one or more service host computers and to an administration client computer, there being a service manager residing on the administration server computer, the system comprising:

a communication connection between the administration client computer

and the administration server computer that can be used for providing a selected user identifier and a corresponding private keyword to the service manager, the user identifier being arranged to identify a user having administrative access to at least one of the services;

an authenticator configured for authenticating the user by comparing the selected user identifier and the corresponding private keyword against a plurality of user identifiers and private keywords stored in a persistent storage area, the comparing performed under control of the service manager;

an access control mechanism for deriving a list of services to which the user associated with the user identifier has administrative access;

a service host verifier for verifying that the user associated with the selected user identifier is permitted to access a selected one of the services in the derived list of services, the verifier residing at the service host computer associated with the selected service and utilizing access control data associated with the selected user identifier in the persistent storage area; and

a data transfer component for transferring one or more management files on the service host computer to the administration server computer thereby facilitating manipulation of the management files utilizing the service manager.

15. A computer readable medium configured to store computer programming instructions for securing access to the administration of a plurality of distinct services residing on one or more service host computers from an administration server computer connected to the one or more service host computers, there being a service manager residing on the administration server computer, the computer readable medium comprising:

computer programming instructions for providing a selected user identi

fier and a corresponding private keyword, the user identifier being arranged to identify a user having administrative access to at least one of the distinct services;

computer programming instructions for authenticating the user by comparing the selected user identifier and the corresponding private keyword against a plurality of user identifiers and private keywords stored in a persistent storage area, the comparing performed under control of the service manager;

computer programming instructions for deriving a list of services to which the user associated with the user identifier has administrative access;

when a request is made to administer a selected one of the services in the derived list of services, computer programming instructions for verifying at the service host computer associated with the selected service that the user associated with the selected user identifier is permitted to access the selected service by examining access control data associated with the selected user identifier in the persistent storage area; and

computer programming instructions for transferring one or more management files on the service host computer to the administration server thereby facilitating manipulation of the management files utilizing the service manager.

3 DETAILED DESCRIPTION OF INVENTION

Background of the Invention

Field of the Invention

The present invention relates generally to computer software and computer network management. More specifically, the present invention relates to server-based management software and software registration in a com

puter network.

Discussion of Related Art

In recent years, computer networks have grown not only in size, such a s number of users or geographical coverage, but also in terms of the typ es of services and protocols a single network can provide and support. Many computer networks allow end users access to all types of services, such as perusing news services or accessing the Internet, and do not res trict users to use mandatory or required network communication protocol.

With the proliferation of services available on some computer networks is the increasing burden on system or network administrators of managin g those services. A system administrator now typically has to install a nd manage software on several servers where each server typically hosts or provides one or more services to network users. Depending on the siz e of the network and the number of services, the day-to-day management, for example, installing, upgrading, and trouble-shooting, the software b ehind these services can become a tedious, error-prone, and time-consumi ng task for a system administrator. This is particularly true with rega rd to system administrators who are not familiar with the network, the s ervers, or the configuration of those servers.

In a large-scale computer network that provides many types of services and applications as described above, there are typically several or many server machines accessible by end-users or clients. The fact that the re are multiple servers on the network is usually transparent to a typic al end-user who is not normally concerned with the physical configuratio n of the network. A system administrator responsible for managing a com puter network normally does so from a server and console, generically de scribed as an administration server, such as a Web server. Figure 1 is a block diagram of a computer network having multiple servers accessible

by end-users and connected to an administration server not configured w ith the automated management capabilities of the present invention. A c ompoter network 102 has an administrator console shown as client 104 con nected to a Web or administrator server 106. Connected to Web server 10 6 are multiple "service" servers 108. From the perspective of administr ation server 106, servers 108 are referred to as management clients. Al though from an end-user's perspective, they are simply servers, where ea ch server may have a particular function or provide a particular service .

When an update, installation, or any type of maintenance is done on ap plication software residing on one of the servers 108 or a new server is added to network 102, the system administrator must modify software on administration server 106 accordingly. For example, if a new feature i s installedon an existing mail server or a new mail server is being add ed, the administrator must note or remember the location and other infor mation of the new feature or server at the time of the update. The admi nistrator installs a new application on a server 110. This information, including the location of any management modules of the new applicatio , which can be in the form of a Uniform Resource Locator, must then be e ntered at console 104. Once manually entered at administrator console 1 04, the information needed to manage the new software or server is refle cted on administrator server 106. At this stage the location of any man agement modules on server 110 are available to the system administrator from administrator console 104. The new mail feature from the example c annot be managed or properly configured by end users until it is "regist ered" with the administrator server 106. Administration server 106 must know where to find the management modules associated with the new mail feature on management clients 108 before end-users can begin using the s oftware.

This is an inefficient process for the administrator and inconvenient for end-users who have come to expect new applications on their networks to be available for use as soon as possible. This process is also error-prone since the administrator has to perform manual or non-automated tasks such as writing down information on the new feature or server during installation, which must later be entered at a administrator console.

This problem is exacerbated if there are dozens of servers, each with many applications (e.g. 35 is not unusual), that have frequent updates, corrections, or new versions that need to be installed in a timely and accurate manner. In this type of setting, managing network services can not only be inefficient, time-consuming, and error-prone, but impractical.

One problem with present Web server based networks typically having multiple service hosts is designing and implementing a user authentication mechanism. A Web server based computer network, or any type of computer network, must have an authentication protocol or mechanism to ensure that a user can perform only those operations or access those files the user is authorized to perform or access. In the case of managing services on the multiple service hosts, there can be more than one system administrator responsible for maintaining the services on these hosts. It is possible that certain administrators are not given complete authorization to perform all possible operations on the Web server and the service hosts, which may only be given to, for example, a senior or "super" system administrator. Thus, since managing services on the hosts is an administration task done through an administration interface, some type of user authentication is necessary.

Although authentication does exist for Web-based networks, present implementations and designs for user authorization are inefficient and repetitive. The authentication referred to here is the verification and aut

horization of system or network administrators for managing services on service hosts in a network from a browser on an administration console.

Typically each service on a service host and its one or more management models have different authentication mechanisms or standards. There is no clear standard on a protocol or process for implementing authentication and access control in a distributed manner on a Web server based system. A system administrator must re-authenticate every time the administrator signs on to a service host since the service hosts are not in communication with each other. A browser program can be run on a client running any type of operating system. thus, the browser being used by the administrator may not be on a UNIX-based client and may not have a known UNIX identity. Since the browser does not have a known UNIX identity, an identity cannot be communicated from one service host to other service hosts. Thus, a system administrator must go through an authentication process for each service host since the administrator does not have a single or globally recognized identity.

Therefore, it would be desirable to manage end-user application software and services available on a computer network from a central location by having any necessary software for managing those applications and services automatically registered at the central location during installation and accessible from a well-known location. It would also be desirable to have an authentication mechanism that provides for single sign on that functions within the environment of a Web server and that server's existing system of user identity and access control. Further, it would be desirable to achieve this from a central location and by assigning a universal identity to a user managing services from a browser in a Web server based network.

Summary of the Invention

To achieve the foregoing, and in accordance with the purpose of the present invention, a method of securing access to a service manager for the administration of services residing on one or more service host computers from an administration server computer is described. In a preferred embodiment of the present invention, a user identifier, such as a user name, and a corresponding password are provided to the service manager, where the user identifier is associated with a system administrator having administrative access to the services. The service manager authenticates the user by comparing the user identifier and password against a list of user identifiers and corresponding passwords stored in persistent memory. A list of services to which the system administrator has administrative access is derived from the data in persistent memory. When the system administrator makes a request to administer one or more services from the list of services, the administrator's access control is verified at the service host computers on which the requested services reside by examining access control data in the persistent memory. Management files are transferred from the service host computers to the administration server computer thereby facilitating manipulation of the management files utilizing the service manager.

In another preferred embodiment, the administration server computer is connected to an administration client computer running a browser program, such as a Web browser. The user identifier and password are provided to the administration server computer over a communications connection between the administration client computer and the administration server computer. The communications connection between the administration server computer and the administration client computer and the connections among the administration server computer and the service host computers use an Internet protocol, such as TCP/IP.

In another aspect of the invention, a system for securing access to a

service manager for administering services on host service computers in a computer network is described. In a preferred embodiment, the service manager resides on an administration server computer connected to multiple host service computers, and is also connected to an administration client computer. A communication connection between the administration client computer and the administration server computer is used for providing a user identifier and password to the service manager. The user identifier represents a user, typically a system administrator, having administrative access to at least one of the services. An authenticator, under the control of a service manager, compares the user identifier and password against a list of user identifiers and passwords stored in persistent memory. An access control mechanism derives a list of services to which the system administrator associated with the identifier and password has administrative access. A service host verifier, residing at the service host computer, verifies that the system administrator is permitted to access the selected services from the list of services by utilizing access control data associated with the system administrator stored in the persistent memory. A data transfer component transfers management files residing on the service host computers to the administration server computer thereby facilitating manipulation of the management files using the service manager.

In another aspect of the present invention, a system for securing administration of services resident on service host computers in a computer network from an administration server connected to an administration client having a browser program and to the service host computers using an Internet protocol, such as TCP/IP, is described. In a preferred embodiment, a user profile data repository stores data relating to user privileges, including a user access level, a list of services, and a password.

A communication interface having a service manager subcomponent resid

ng on the administration server accepts a user name and password and pa sses the information to the user profile data repository. A component c onfiguration directory that can reside on a service host contains compon ent configuration files that store management modules belonging to servi ces. The management modules contain management data that can be used in administering the services. The communication interface also has a ser vice host subcomponent that resides on the administration server compute r that accepts the user name and password and passes the information the service host computers for verification at the service hosts by examini ng data relating to user privileges stored in the user profile data repo sitory.

The invention, together with further advantages thereof, may best be u nderstood by reference to the following description taken in conjunction with the accompanying drawings.

Detailed Description of the Preferred Embodiments

Reference will now be made in detail to a preferred embodiment of the invention. An example of the preferred embodiment is illustrated in the accompanying drawings. While the invention will be described in conjun ction with a preferred embodiment, it will be understood that it is not intended to limit the invention to one preferred embodiment. To the con trary, it is intended to cover alternatives, modifications, and equivale nts as may be included within the spirit and scope of the invention as d efined by the appended claims.

A method and system for managing software applications and services fr om a central location in a computer network is described in the various drawings. In a large-scale computer network having multiple servers and a large end-user base, managing applications and software on the networ k is a time-consuming and error-prone task. Typically, a system adminis

trator installs a new application or service on a service host, i.e., on e of the network servers which is normally done at the server. Informat ion relating to management of the application, in particular the locatio n and names of files of management modules, are manually noted by the sy stem administrator. This information is then entered on an administrato r server through an administrator console. Once the location of the new application management module is known to the administrator server, for example a Web server, end users can access the new application. This p rocess becomes cumbersome and inefficient when there are many servers on the network, each having many applications that require frequent updati g, modifying or replacing. This problem is particularly acute from the end-user's perspective in that the expectation that an application be av ailable for use soon after it is received is high. The non-automated tw o-step process described increases the time before an application can be available to users on the network.

The present invention is a method of automating the process of registe ring new applications and services at a central management location, su ch as a Web server, thereby reducing the amount of information the syste m administrator must remember and making a service available to end-user s sooner. In the described embodiment, the present invention involves h aving a management console program residing on an administration server that manages other servers or service hosts on the network, also referre d to as management clients in the sense that these servers are "clients" of the administration server. The described embodiment also includes a persistent storage area containing a database for storing management in formation and uses (e.g. system or network administrators) authenticatio n information relating to the services or the service hosts and a "well-known" directory associated with each management client. In other prefe rred embodiments, described in more detail below, the storage areas, for

example, can be distributed over the network instead of being associated with only one server. In another preferred embodiment, the management console program does not reside entirely on the administration server, but can also be distributed between the server and an administrator client machine. These components are shown in Figure 3.

Figure 2 is a block diagram of server side components of a computer network in accordance with one embodiment of the present invention. A server-side configuration 200 of a complete network (not shown) can be viewed as having two sections, a section 202 representing an administration side and a section 204 representing network servers, or service hosts. Not shown in Figure 2 are the network end users on client machines which can typically access network servers 206 to provide services or for running applications, or performing other network operations. Although the end-users of a computer network are one of the beneficiaries of the present invention in that services and applications on the network are available to them sooner and do not go down as often, in the described embodiment the invention is used by a system administrator or network manager (i.e., the user).

In the described embodiment, management clients 206 are managed through a Web server 208. In other preferred embodiments, server 208 can be another type of server, such as a more generic administration server, or be a server that has other functions depending on the size of the network and the capacity of the server. In any case, server 208 in the network has the role of managing management clients 206. One feature of server 208 is that it contains a management console program 210, described in greater detail below. Another feature of Web server 208 is that it has access to a persistent storage area database 212 that stores service management module information. Web server 208 communicates with storage 212 through the light-weight directory access protocol (LDAP) 214. In ot

ber preferred embodiments, other data access protocols can be used betwe
en server 208 and storage area 212. Storage area 212 is also accessible
by management clients 206. Persistent storage 212 is a reliable databa
se that stores data, in the described embodiment, in a hierarchical form
at. In other preferred embodiments, the database can be in relational d
atabase format or store data in an object-oriented type data repository.

In addition, in other preferred embodiments, storage 212 can be distri
buted across persistent storage areas part of management clients 206, We
b server 208, and other persistent storage mediums available to the netw
ork and accessible by the servers.

As mentioned, the present invention is used primarily by a system admi
nistrator. The administrator accesses server 208 through a special clie
nt administrator console 216. In the described embodiment, console 216
is equipped with a Web-based browser program that allows the administrat
or to access server 208 and, more specifically, use management console p
rogram 210 and storage area 212. Server 208 can also be referred to as
a management console host from the perspective of browser host 216. As
will be described in greater detail below, a system administrator can us
e browser host 216 to manage software applications and services on manag
ement clients 206.

Management clients 206 can include all or some of the servers on the n
etwork. Those that are managed by a system administrator through Web se
rver 208 communicate with storage 212 via LDAP. Each management client
has one or more services shown at 218 and one or more corresponding mana
gement modules shown at 220 on service host 207. When a new service is
installed or an existing service is upgraded, an entry in management mod
ule area 220 is altered. As described in greater detail below, this alt
eration is reflected in corresponding entries in persistent storage 212.

Although services 218 are shown separately from management modules 220

in Figure 2, the two components are integral to each other. In other words, a service's management module is integrally bound with the main body or functional modules of the service. However, the two components still have separate roles. Management modules 220 are stored in configuration files, a configuration component directory is described in greater detail below. In other preferred embodiments the information in management modules 220 can be stored in other formats such as a database or a standard directory that also contains other non-management files.

The remaining components in Figure 2 relating to the management console program address authentication and access control features. Management console program 210 has an authentication layer 222 that performs user verification and authorization functions described in greater detail with regard to Figures 8 to 11 below. Associated with console host 208 is a Common Gateway Interface, or CGI program, used by a Web server to execute programs. In the described embodiment, a CGI program 224 is used to execute programs from console host 208 and is logically divided in to two parts: a management console CGI 226 and a servlet CGI 228. Management console CGI 226 communicates with management console program 208 and is discussed in greater detail with respect to Figures 9 and 10. Servlet CGI 228 communicates authentication data from console host 208 to the service hosts 206, and is a component well known in the art.

On service hosts 206 is a corresponding authentication and access control layer 230 that is part of management module component 220. Authentication layer 230 receives data from console host 208 through servlet CGI 228. These components are used to ensure that a system administrator logging on to use the management console program to manage particular services is authorized to manage those services and also allows a "super" system administrator to add and delete administrators and particular privileges in the management console framework. In the described embodiment

, this functionality is illustrated through a graphical user interface s hown in Figure 8. Service hosts 260 re-authenticate a user's access con trol and authorization with persistent data storage 270.

Figure 3 is a flowchart showing an overview of a process for registeri ng a new service on a network in accordance with one embodiment of the p resent invention. The flowchart shows the steps taken by a system admin istrator when registering either a new service, upgrading a service, or adding a new management client to the network. At step 302 a service is installed on a particular management client. This is typically done th rough a client machine functioning as a browser host and is usually perf ormed by a system administrator. A management module, associated with t he service, is a segment of executable code that is also installed on th e management client. An example of a management module on a mail server is a module indicating a maximum quota per end-user; that is, the maxim um amount of memory a user can take up. Another example is a Web server owned by an ISP (Internet service provider) that hosts web sites for it s customers. In this context a management module can manage the additio n of a new Web site on the Web server.

The management module can be one of several types. In the described e mbodiment, the types of management modules are browser-based, X-based, a nd command line. A browser-based management module is associated with a n application that is executed in a Web browser. It is anticipated that a large majority of the application types will be applications that run in a Web browser. An X-based management module is typically associated with a stand alone application that is run based on the X-protocol, a c omponent of the UNIX operating system. These applications are generally not run from within a browser but from the operating system shell. It is derived from standard and well-known X-windows, a UNIX-based graphica l user interface. A command line management module is associated with a

n application which is managed using command lines, but can be embedded and executed from a Web browser. A command line may or may not have run time parameters as is described below. Examples of command line command s are "ls" (obtain a list of files), "whoami" (return information on cur rent user), and "ps" (provide information on performance states). In oth er preferred embodiments other types of management modules can be instal led.

At step 304 the system administrator registers the service and managem ent modules on the management client. In the described embodiment this is done by running a command referred to as mc_reg on the management cli ent. By registering the service and management modules, the administrat ion server (server 208 in Figure 2) is informed of what type of module i s being installed. Typically, a system administrator registers several new services on various management clients. Thus, steps 302 and 304 are repeated for several services on various management clients. Once a se rvice is registered on a service host, certain files referred to as comp onent configuration files storing management data are created and stored in a component configuration directory on the service host. Step 304 i s described in greater detail with respect to Figure 4.

At step 306 a "discover" routine is initiated through a graphical user associated interface associated with management console program 210 and is run on a service host. This routine allows the management console p rogram to register a particular service host. The system administrator, for example through browser host 216, instructs the management console to go to a particular service host or group of service hosts and check t o see what has been registered. In the described embodiment this is don e by the management console by checking a well-known directory referred to as the component configuration directory on the service hosts indicat ed by the system administrator. Step 306 is described in greater detail

in Figure 5. In a preferred embodiment the discover routine can be run locally on the service host at the time the service is being installed at step 302. The service host can then broadcast the results of the remote or auto discover to the management console program. In the described embodiment, the system administrator can tell the management console to go register all the service hosts that were recently modified, upgraded, or newly added by the administrator. In the described embodiment, the management console program proceeds to check those service hosts and will register any updates by checking the component configuration directory. Once all the modified service hosts have been registered, end users can begin using the services or applications and the registration process is complete.

Figure 4 is a flowchart showing in greater detail step 304 of Figure 3 of registering a service in accordance with one embodiment of the present invention. Step 304 introduced the process of registering a new service on a service host so that the management console can later discover that the a new service has been registered on that host as instructed by a system administrator. At step 402 the service or application type is identified to the service host. As described above, in the described embodiment, a service can be one of three types: browser-based, X based, and command line. In other preferred embodiments, additional types can be entered. In the described embodiment, this step is performed on the service host and is one way of informing the management console of the application type. In other preferred embodiments, this information can be entered at the browser host. Information inputted at the service host after step 402 depends on the type of service identified. If the service is Web-based, the flowchart proceeds with step 404. At step 404 the system administrator enters the location of the service's management module on the service host. In the case of Web-based services, the locatio

n is typically in the form of a Uniform Resource Locator. At step 406 the service type and the URL of the management entry are saved as parameters in a well-known location on the service host. In the described embodiment, these two items of information, referred to as components, are saved in a UNIX file referred to as a component configuration file in the directory referred to as a component configuration directory. In other preferred embodiments, other directories on the service host can be used to store these components.

At step 408 the two components contained in a service management entry are assigned component identifiers. In the described embodiment, this consists of two parts: (1) a unique identifier (such as a Solaris package name, e.g. SUNWFTP), and (2) a version number. Thus, the URL and the service type components are assigned a component identifier and saved in a file in the component configuration directory. In addition a "user friendly" name for the service, which up to this point has been a unique but lengthy and cryptic name, is entered. This user friendly name is the name that will be displayed on the graphical user interface, described in greater detail with respect to Figure 6 below. At step 420 the data or components described in steps 406 and 408 are stored in an appropriate file in the component configuration directory. Thus, after step 420 all the information needed to perform step 306 of Figure 3 (the "discovery" process) for a Web-based type service is stored in an appropriate file at a well-known directory and the process is complete.

Returning to step 402, if the service type is X-based, control proceeds with step 410. As described above, an X-based type service is typically associated with a stand alone application that is run based on the X protocol, a component of the UNIX operating system. At step 410, the system administrator enters the path necessary to invoke the X-based application. At step 412 a UNIX user and user group are entered in order to

invoke the X-based application. Control then goes to step 408 where the path, user name, and group are assigned component identifiers. At step 420 the component identifiers are stored in an appropriate file in the component configuration directory.

For command line type management modules, the system administrator enters data similar to the X-based type: a path to invoke the command line, and a UNIX user and group name necessary to invoke the UNIX application, as shown at step 414. At step 416 the system administrator determines whether there are any runtime parameters in the command (reflected in the command line type management module). These parameters are not entered at the time the service is being registered but at the time the command is executed or run by the end-user. The graphical user interface is modified or customized to reflect whether the end-user can enter runtime parameters (e.g., options the user can select at the time the service is being used). If there are runtime parameters, the system administrator supplies them in response to a prompt from the management console's graphical user interface. At step 418 the system administrator enters static parameters required by the command. A command line type management module will always have static parameters regardless of whether the command has runtime parameters. Control then goes to step 408 where all the data is assigned component identifiers, as was done for X-based and Web-based management modules. The component identifiers are then saved in files stored in the configuration component directory at step 420. In the described embodiment, the file name has the format of "component identifier - version number" which facilitates determining the number of components that are registered in the directory where each component has one file. In other preferred embodiments, the file name can be in another formats where there is one file per command, e.g. component identifier command #. .

Figure 5 is a flowchart showing in greater detail step 506 of Figure 3 in accordance with one embodiment of the present invention. In the described embodiment, a service host has a component software segment containing that contains all the management modules of the services on that service host. The component configuration directory resides in this segment. The service host also has a management console framework segment that contains code also contained in the management console program residing on the administration server. For example, the mc_reg command and ISP remote shell code, a program for remotely executing X based and command line management programs, reside in both the management console and the service host. Figure 5 describes a discovery process that searches the component software segment on a service host for management modules that have not yet been registered using software in the management console framework segment.

At step 502 a system administrator specifies a service host name or a service name through a graphical user interface on the browser host. Examples of graphical user interfaces used in the described embodiment are shown in greater detail in Figures 6, 7, and 8. As described above, there can be many service hosts, each of which have several services available. These choices are presented to a system administrator through a user interface. Typically an administrator will choose all the service hosts that contain services that were recently modified or added, and will enter all these service hosts at once from the browser host. At step 504 the management console host connects to the one or more service hosts specified at step 502 to scan a well-known directory for component configuration files. In the described embodiment the well-known directory is the component configuration directory. The management console communicates with the service host through a standard CGI (Common Gateway Interface) program, typically used to initiate a Web-based program from a Web

server, and is well-known in the art. In other preferred embodiments t he CGI program may not be needed if the administration server is not a W eb-based server. The scanning is performed using a command line program that sends commands across a network connection and have them executed on the destination server. More specifically, in the described embodime nt, the commands are executed by the management console, over the networ k connection, on the service host. In the described embodiment, this is done using an ISP remote shell protocol. Thus, during the scan the UNI X "list files" command, ls, is executed in the component configuration d irectory to get a list of the component configuration files. A list of files that need to be registered with the management console is sent to the administration server.

At step 506 the management console examines the list of files "discove red" on all the service hosts that were specified in step 502. The same connection between the management console and the service hosts is then used to retrieve the contents of these files. In the described embodim ent, the UNIX "concatenate" command, cat, is used on the service host to retrieve the content of each file. In other preferred embodiments, sim ilar commands for retrieving the content of a file in other operating sy stems can be used. Once the contents of each file to be registered are retrieved from the service hosts, the content of each individual file is parsed using standard and well-known parsing techniques by the managemen nt console on the administration server. In the described embodiment, a component configuration file is flat ASCII file. By parsing the conten t of a file, the file's user friendly name, component identifiers, and o ther command execution information are identified for each file. In the described embodiment, this information reflects the information that wa s saved in the component configuration directory for each of the three m anagement module types as shown in Figure 4.

At step 508 the data parsed from the component configuration files is stored on a persistent storage area. As described above, a component configuration file contains all the information that is needed to launch a corresponding service. This information is now stored in a database on persistent storage accessible by the management console program and by the service hosts. A system administrator can now manage a service through the management console by modifying the content of that service's management data stored in the persistent and reliable database. In the described embodiment, data on the persistent storage remains when the network is down or when the management console is not active, and is accessible through the light-weight directory access protocol (LDAP). In other preferred embodiments, alternative access protocols can be used depending on the type of storage being used and the network.

Figures 6 and 7 are screen shots of a graphical user interface displayed on the browser host in accordance with one embodiment of the present invention. Figure 6 is an initial screen shot of the "Register Services" user interface. A window 602 contains a text entry sub window 604 in which a system administrator enters the name of a service host on which services the administrator wants to manages reside. In the described embodiment there is an area to enter one service host. In other preferred embodiments an administrator can enter more than one service host. Also shown in text entry sub-window 606 in which an administrator can enter a service host name that contains services the administrator wants to unregister. Once the choices have been entered, the user can click on button 608 to retrieve a list of services that the user is authorized to manage on that service host. The administrator can also press button 610 to retrieve a list of services on that service host which can be unregistered.

Figure 7 is a screen shot showing another segment of the "Register Ser

vices" user interface. This graphical user interface allows a system administrater to select services that the administrator is authorized to manage. User authorization and access control is described in greater detail below. A list of services 612 is displayed in a window 614. List 612 is derived from data relating to the user stored in the database and contains those services available on the service host entered in field 604 of Figure 6. The system administrater selects those services he wants to manage or access. In the described embodiment this is shown with an asterisk to the left of the service name, such as the Sun News (TM) service 616. Once the service or services have been selected, the user clicks on the "Register Services Selected Above" bar 618. In the described embodiment this is done using a pointing device such as a mouse or track ball and is implemented in a window environment. In other preferred embodiments, a non-graphical user interface, such as a simple text based interface or a more sophisticated voice-recognition based interface can be used to enter this information, as well as the information described below with respect to the other screens.

As described above, a management console program of the present invention includes a "single sign-on" method of user authentication and access control that benefit from having a central management console for managing services on multiple service hosts in a distributed Web-based network. Presently in Web-based networks a system administrater responsible for maintaining services available on multiple service hosts must re-authenticate and pass the administrater's credentials to each service host to which the administrater logs on. This is true since the administrator, operating from a browser, does not have a single, universal identity that can be used for authentication. Here authentication refers to verifying credentials and authorizations of a user before being allowed to manage a particular service host or, more specifically, perform operations

for managing services on a particular service host. It is necessary to have a consistent understanding throughout the network of who the user is and what that user is allowed to do on the service hosts.

The present invention allows centralized management and user single sign on for authentication relating to management of services on service hosts from a browser host. The management console program 210 of Figure 2 contains an authorization and access control component or layer 222. This authorization layer accesses user data from database 212 for verification and communicates this information to corresponding authorization or authentication layers 320 on a service host 250. The information is handled and transmitted to each service host a system administrator wants to manage, without having the administrator re-authenticate on each individual service host.

Information relating to each user is stored in database 212 and information entered by a user is authenticated against this information. The information, or credentials, if verified, is passed through a CGI program to the service hosts indicated by the user. Once received by the service hosts the information is re-authenticated against the user profile into the database on behalf of the system administrator; in other words, this is done "behind the scenes" without intervention or any extra steps from the user. The user only has to log on (i.e. enter certain information such as name and password) to the management console through a browser once and this information is passed on to the service hosts automatically.

Figure 8 is a screen shot of a graphical user interface relating to the access control and authentication of a user of the management console program in accordance with one embodiment of the present invention. A window 702 has the heading "Manage Administrators." This window is used to enter new administrators and associated passwords and services the ad

w administrator will be allowed to manage. Within window 702 is a sub-w indow 704 for entering an administrator name and sub-windows 706 and 708 for entering and re-entering a password. In the lower portion of windo w 702, another sub-window 710 contains a list of services from which the administrator entered in sub-window 704 will be allowed to manage. Onc e the services are selected by the managing or "super" administrator, th e button 712 is pressed.

Figures 9 and 10 are flowcharts of a process for enforcing access cont rol and authorization in the management control program in accordance wi th one embodiment of the present invention. The enforcement process beg ins with a user pointing the browser host (i.e. administration console 2 16 of Figure 2) to a URL of the management console host. Thus, at step 802 the user enters the URL of the console host from the browser host. The URL for the management console is in the form of a standard URL in a Web-based network. In other preferred embodiments, other types of loca tors can be used depending on the type of network.

At step 804 the administrator/user is challenged for a user name and p assword for access to the management console program on the console host . At step 806 the management console accepts the user name and password entered in step 804 and the user is authenticated. This step is describ ed in greater detail in Figure 11. The management console displays the services on a selected service host as shown in area 612 of Figure 6 tha t the user is authorized to manage by examining data in database 212. T his is done by using the management console segment of the CGI as shown in Figure 2. In the described embodiment, an administrator's authorizat ion is defined in terms of services that the administrator is allowed to manage. During this step the management console constructs a URL for e ach service and host that the administrator is allowed to manage. This process is also described in greater detail with respect to Figure 11.

The URLs allows the console host to locate each service host and service that can be managed by the administrator.

At step 808 the user selects an instance of a service (i.e. a particular service from a service host) that the user wants to manage. A service can reside on several different service hosts so the user must choose an instance of a service from a particular service host. By selecting the user friendly name the user has selected one of the URLs constructed in step 806. At step 810 the management console host initiates the servlet CGI component of the CGI. In the described embodiment, this is done by comparing the user credential's or profile against the user's authentication and access control data in the database. This verification is performed before a connection is made to the service host by servlet CGI 224 as an extra precaution against users trying to manage services on that service host without going through management console host 208. Since this is a network environment, it is possible for a user to bypass the console host verification steps and attempt to access services on a service host directly from a client machine, instead of from browser host 216 of Figure 2. Thus, the user credentials are compared against the user data stored in database 212 by the servlet CGI.

At step 812 the servlet CGI uses a standard procedure for passing the user credentials to the service host or hosts indicated by the user. In the described embodiment, once the data is received, the service host performs authentication and access control using the data by comparing it against data in the database. In other preferred embodiments, this step may not be necessary depending on independent security features available on the particular network implementing the management console program. This re-authentication is done without any intervention from the user and is performed to ensure that a user is not attempting to log on directly to the service host thereby circumventing the authentication and a

ccess control layer of the management console host. This is by performing a second check against the database without requiring the user to perform any extra operations. the management console can centralize management of services in the network. If the re-authentication is successful at step 814, management console program on the console host allows the user to perform management operations on the selected service or services from the browser as shown at step 816 at which point the enforcement process is complete. If the re-authentication is not successful, the user is denied authority to manage the selected service and is shown the login screen again.

Figure 11 is a flowchart showing in greater detail step 806 of Figure 9. In step 806 the user is authenticated and the services that the user is authorized to access are determined and the URLs to each of those services are constructed. At step 902 the management console host authenticates the user by retrieving information relating to the user from the database. This information consists of the user's name and password. Once the user name and password are verified, a list of services that the user is autherized to manage is derived. At step 904 the console host initiates the management console segment 226 of the CGI program with the user credentials which were verified at step 902. As described above, this is the first step in establishing a link with a service host.

The other component of the CGI is the servlet CGI (item 224 of Figure 2) is used to establish the connection with the service host. At step 906 the management console CGI queries database 212 of Figure 2 to obtain the list of services the user is authorized to manage. Links to these services are constructed in the form of URLs to all the services on the list. The database contains an entry for each user that contains information including the user's name, password, level (e.g. super system administrator), and a list of services that the user is allowed to manage.

A super system administrator can manage all services and define access control parameters for the other users (e.g. proxy system administrators). The list of services contains "user friendly" names of the services (also contained in the database) instead of the services URL. Control then returns to step 806 of Figure 8 where the user selects which service he wants to manage from the list of services.

The present invention employs various computer-implemented operations involving data stored in computer systems. These operations include, but are not limited to, those requiring physical manipulation of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. The operations described herein that form part of the invention are useful machine operations. The manipulations performed are often referred to in terms, such as, producing, identifying, running, determining, comparing, executing, downloading, or detecting. It is sometimes convenient, principally for reasons of common usage, to refer to these electrical or magnetic signals as bits, values, elements, variables, characters, data, or the like. It should remembered, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities.

The present invention also relates to a device, system or apparatus, such as browser host 216 and management console host 208 for performing the aforementioned operations. The system may be specially constructed for the required purposes, or it may be a general purpose computer selectively activated or configured by a computer program stored in the computer. The processes presented above are not inherently related to any particular computer or other computing apparatus. In particular, various general purpose computers may be used with programs written in accordance w

ith the teachings herein. or, alternatively, it may be more convenient t
o construct a more specialized computer system to perform the required o
perations.

Figure 12 is a block diagram of a general purpose computer system 1000
suitable for carrying out the processing in accordance with one embodim
ent of the present invention. The management console program including
the authentication and access control layer can resides on such a genera
l purpose computer. In addition, browser host 216 can be such a general
purpose computer. Figure 12 illustrates one embodiment of a general pu
rpose computer system. Other computer system architectures and configur
ations can be used for carrying out the processing of the present invent
ion. Computer system 1000, made up of various subsystems described belo
w, includes at least one microprocessor subsystem (also referred to as a
central processing unit, or CPU) 1002. That is, CPU 1002 can be implem
ented by a single-chip processor or by multiple processors. CPU 1002 is
a general purpose digital processor which controls the operation of the
computer system 1000. Using instructions retrieved from memory, the CP
U 1002 controls the reception and manipulation of input data, and the ou
tput and display of data on output devices.

CPU 1002 is coupled bi-directionally with a first primary storage 1004
, typically a random access memory (RAM), and uni-directionally with a s
econd primary storage area 1006, typically a read-only memory (ROM), via
a memory bus 1008. As is well known in the art, primary storage 1004 c
an be used as a general storage area and as scratch-pad memory, and can
also be used to store input data and processed data. It can also store
programming instructions and data, for example in the form of a hierarch
ical database such as database 212 in addition to other data and instruc
tions for processes operating on CPU 1002, and is used typically used fo
r fast transfer of data and instructions in a bi-directional manner over

the memory bus 1005. Also as well known in the art, primary storage 10
06 typically includes basic operating instructions, program code, data a
nd objects used by the CPU 1002 to perform its functions. Primary stora
ge devices 1004 and 1006 may include any suitable computer-readable sto
rage media, described below, depending on whether, for example, data acce
ss needs to be bi-directional or uni-directional. CPU 1002 can also dir
ectly and very rapidly retrieve and store frequently needed data in a ca
che memory 1010.

A removable mass storage device 1012 provides additional data storage
capacity for the computer system 1000, and is coupled either bi-directio
nally or uni-directionally to CPU 1002 via a peripheral bus 1014. For e
xample, a specific removable mass storage device commonly known as a CD-
ROM typically passes data uni-directionally to the CPU 1002, whereas a f
loppy disk can pass data bi-directionally to the CPU 1002. Storage 1012
may also include computer-readable media such as magnetic tape, flash m
emory, signals embodied on a carrier wave, PC-CARDS, portable mass stora
ge devices, holographic storage devices, and other storage devices. A fi
xed mass storage 1016 also provides additional data storage capacity and
is coupled bi-directionally to CPU 1002 via peripheral bus 1014. The mo
st common example of mass storage 1016 is a hard disk drive. Generally,
access to these media is slower than access to primary storage 1004 and
1006. Mass storage 1012 and 1016 generally store additional programming
instructions, data, and the like that typically are not in active use b
y the CPU 1002. It will be appreciated that the information retained wi
thin mass storage 1012 and 1016 may be incorporated, if needed, in stand
ard fashion as part of primary storage 1004 (e.g. RAM) as virtual memory

In addition to providing CPU 1002 access to storage subsystems, the pe
ripheral bus 1014 is used to provide access other subsystems and devices

as well. In the described embodiment, these include a display monitor 1018 and adapter 1020, a printer device 1022, a network interface 1024, an auxiliary input/output device interface 1026, a sound card 1028 and speakers 1030, and other subsystems as needed.

The network interface 1024 allows CPU 1002 to be coupled to another computer, computer network, or telecommunications network using a network connection as shown. Through the network interface 1024, it is contemplated that the CPU 1002 might receive information, e.g., data objects or program instructions, from another network, or might output information to another network in the course of performing the above-described method steps. Information, often represented as a sequence of instructions to be executed on a CPU, may be received from and outputted to another network, for example, in the form of a computer data signal embodied in a carrier wave. An interface card or similar device and appropriate software implemented by CPU 1002 can be used to connect the computer system 1000 to an external network and transfer data according to standard protocols. That is, method embodiments of the present invention may execute solely upon CPU 1002, or may be performed across a network such as the Internet, intranet networks, or local area networks, in conjunction with a remote CPU that shares a portion of the processing. Additional mass storage devices (not shown) may also be connected to CPU 1002 through network interface 1024.

Auxiliary I/O device interface 1026 represents general and customized interfaces that allow the CPU 1002 to send and, more typically, receive data from other devices such as microphones, touch-sensitive displays, transducer card readers, tape readers, voice or handwriting recognizers, biometrics readers, cameras, portable mass storage devices, and other computers.

Also coupled to the CPU 1002 is a keyboard controller 1032 via a local

bus 1034 for receiving input from a keyboard 1036 or a pointer device 1 038, and sending decoded symbols from the keyboard 1036 or pointer devic e 1038 to the CPU 1002. The pointer device may be a mouse, stylus, trac k ball, or tablet, and is useful for interacting with a graphical user i nterface.

In addition, embodiments of the present invention further relate to co mputer storage products with a computer readable medium that contain pro gram code for performing various computer-implemented operations. The c omputer-readable medium is any data storage device that can store data w hich can thereafter be read by a computer system. The media and program code may be those specially designed and constructed for the purposes of the present invention, or they may be of the kind well known to those o f ordinary skill in the computer software arts. Examples of computer-re adable media include, but are not limited to, all the media mentioned ab ove: magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD-ROM disks; magneto-optical media such as flopt ical disks; and specially configured hardware devices such as applicatio n-specific integrated circuits (ASICs), programmable logic devices (PLDs), and ROM and RAM devices. The computer-readable medium can also be di stributed as a data signal embodied in a carrier wave over a network of coupled computer systems so that the computer-readable code is stored an d executed in a distributed fashion. Examples of program code include b oth machine code, as produced, for example, by a compiler, or files cont aining higher level code that may be executed using an interpreter.

It will be appreciated by those skilled in the art that the above desc ribed hardware and software elements are of standard design and construc tion. Other computer systems suitable for use with the invention may inc lude additional or fewer subsystems. In addition, memory bus 1006, perip heral bus 1014, and local bus 1034 are illustrative of any interconnecti

on scheme serving to link the subsystems. For example, a local bus could be used to connect the CPU to fixed mass storage 1016 and display adapter 1020. The computer system shown in Figure 12 is but an example of a computer system suitable for use with the invention. Other computer architectures having different configurations of subsystems may also be utilized.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. Furthermore, it should be noted that there are alternative ways of implementing both the process and apparatus of the present invention. For example, although the invention has been described using a Web server as the administration server, a non-Web based server can also be used to run the management console program. In another example, database 212 can be a distributed database stored on the console host and various service hosts rather than at a single persistent database. In yet another example, data retrieval protocols other than LDAP can be used to retrieve data from database 212 or from a "flat file stored on a persistent storage area. In yet another example, the discover routine can be run "locally" on a service host while the service is being installed instead of at a later time on the console host. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

4 BRIEF DESCRIPTION OF DRAWINGS

Figure 1 is a block diagram of a computer network having multiple servers accessible by end-users and connected to an administration server not configured with the automated management capabilities of the present i

ovention.

Figure 2 is a block diagram of server side components of a computer ne twork in accordance with one embodiment of the present invention.

Figure 3 is a flowchart showing an overview of a process for registeri ng a new service on a network in accordance with one embodiment of the p resent invention.

Figure 4 is a flowchart showing in greater detail, step 304 of Figure 3 of registering a service in accordance with one embodiment of the prese nt invention.

Figure 5 is a flowchart showing in greater detail step 306 of Figure 3 in accordance with one embodiment of the present invention.

Figure 6 is a screen shot of a graphical user interface displayed on t he browser host in accordance with one embodiment of the present inventi on.

Figure 7 is a screen shot of a graphical user interface displayed on t he browser host in accordance with one embodiment of the present inventi on.

Figure 8 is a screen shot of a graphical user interface relating to th e access control and authentication of a user of the management console program in accordance with one embodiment of the present invention.

Figure 9 is a flowchart of a process for enforcing access control and authorization in the management control program in accordance with one e mbodiment of the present invention.

Figure 10 is a flowchart of a process for enforcing access control and authorization in the management control program in accordance with one embodiment of the present invention.

Figure 11 is a flowchart showing in greater detail step 906 of Figure 9.

Figure 12 is a block diagram of a typical computer system suitable for

implementing an embodiment of the present invention.

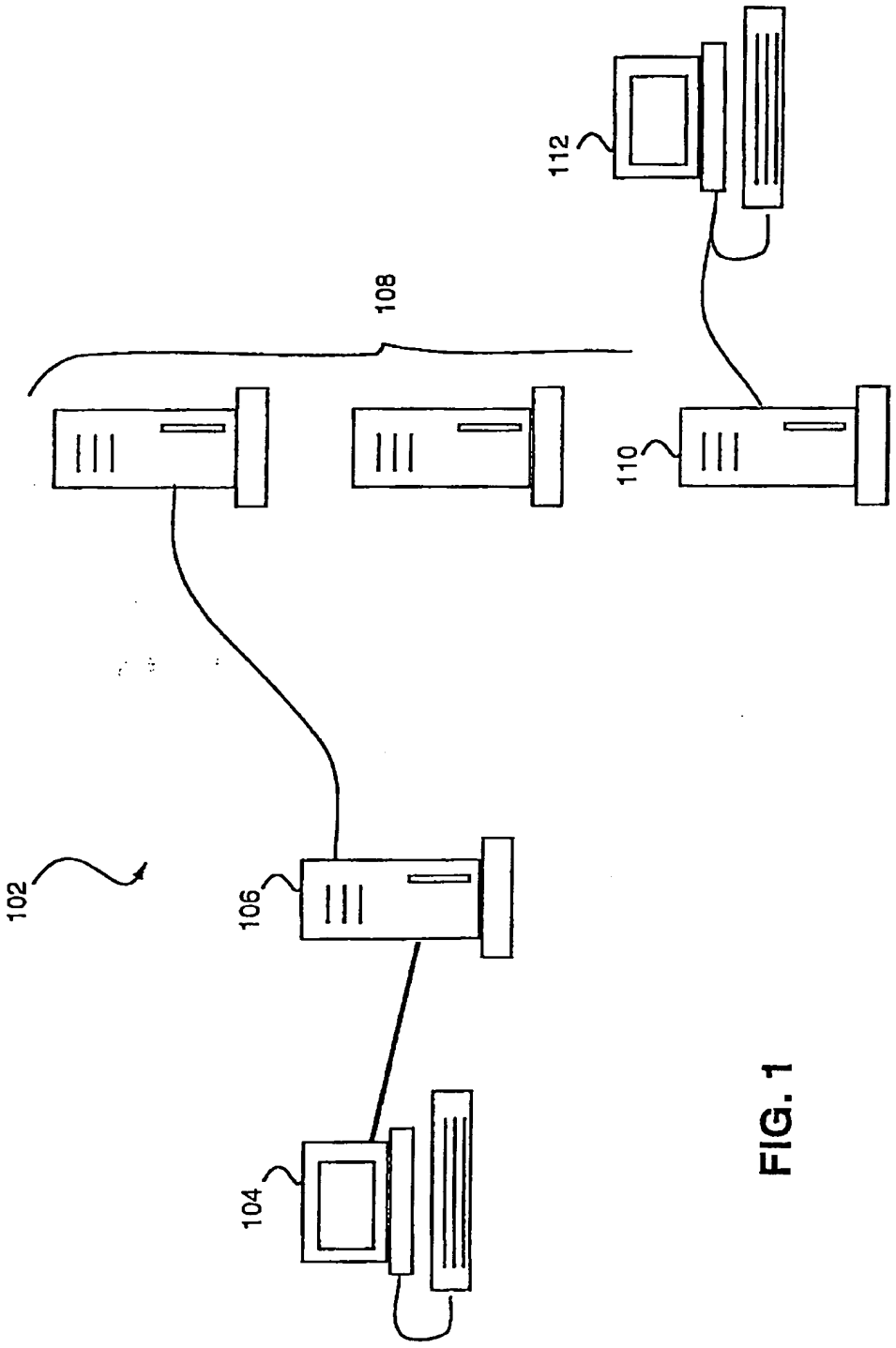


FIG. 1

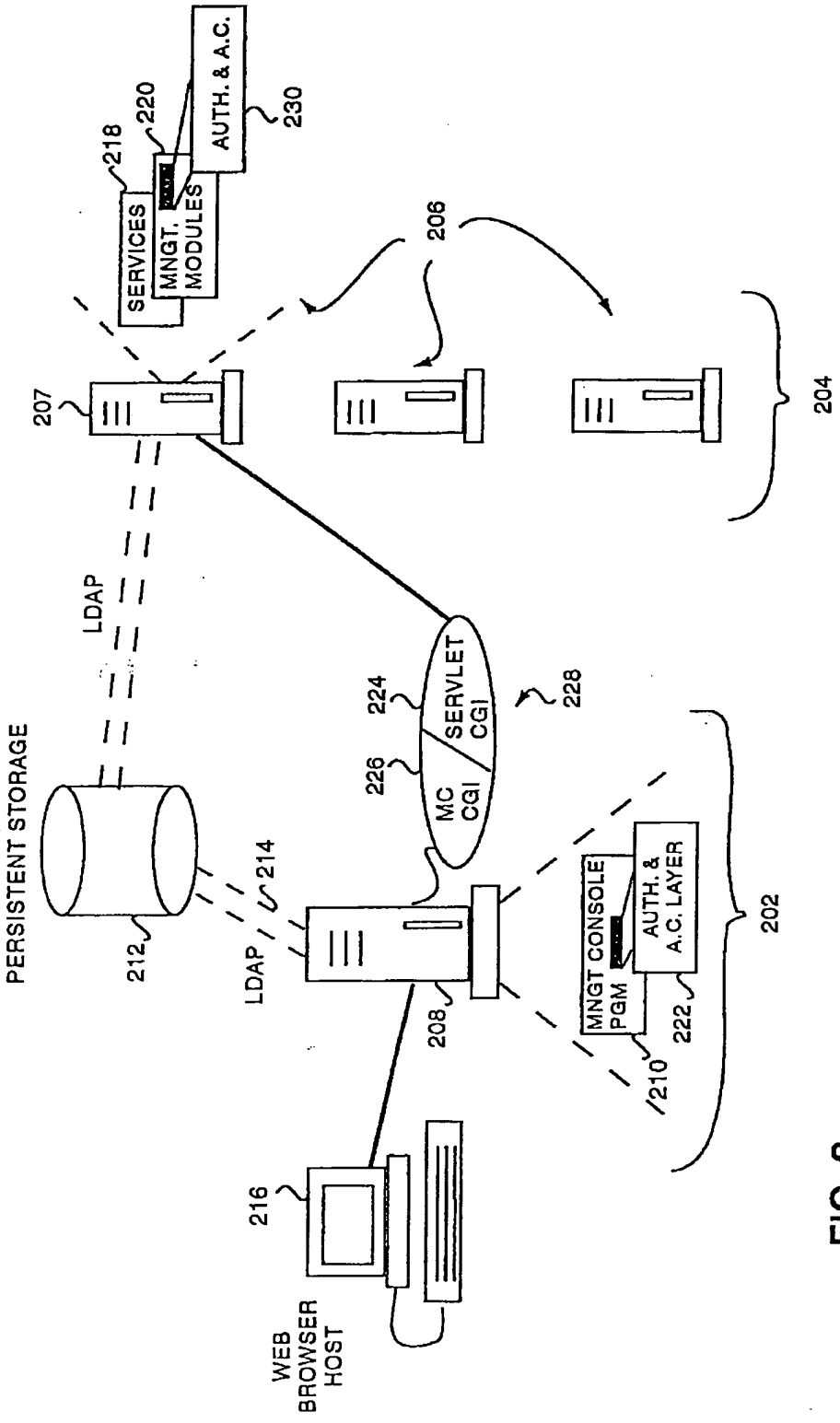

AUTH. & A.C.
230
220
218
SERVICES
MNGT. MODULES
206
207
204
LDAP
DATABASE/ PERSISTENT STORAGE
212
214
LDAP
224
SERVLET CGI
228
226
MC CGI
MNGT CONSOLE PGM
AUTH. & A.C. LAYER
202
208
222
210
216
WEB BROWSER HOST

FIG. 2

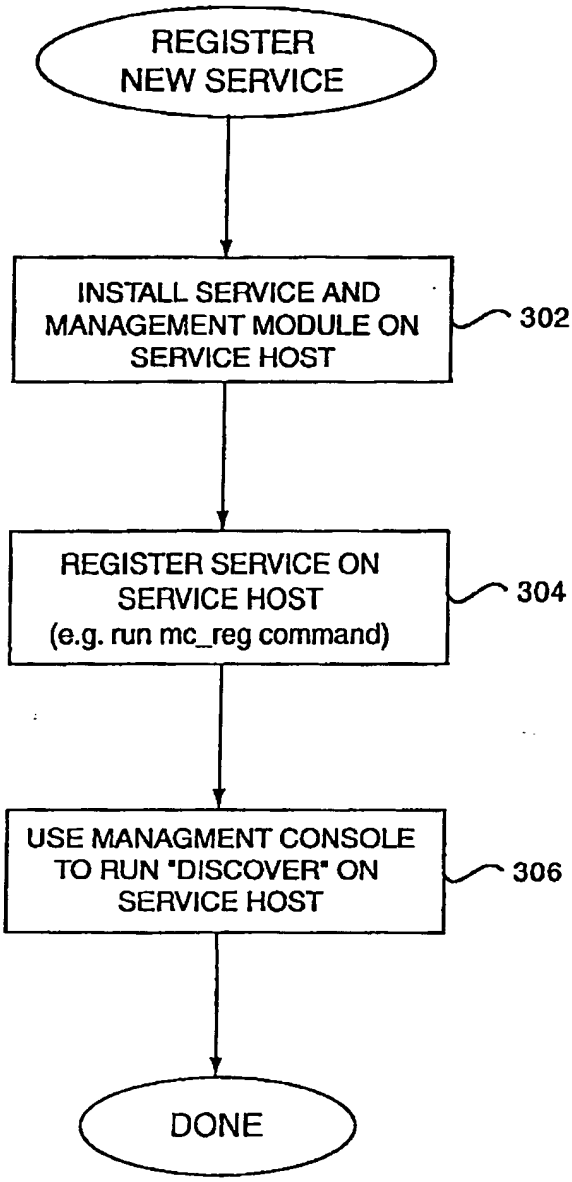
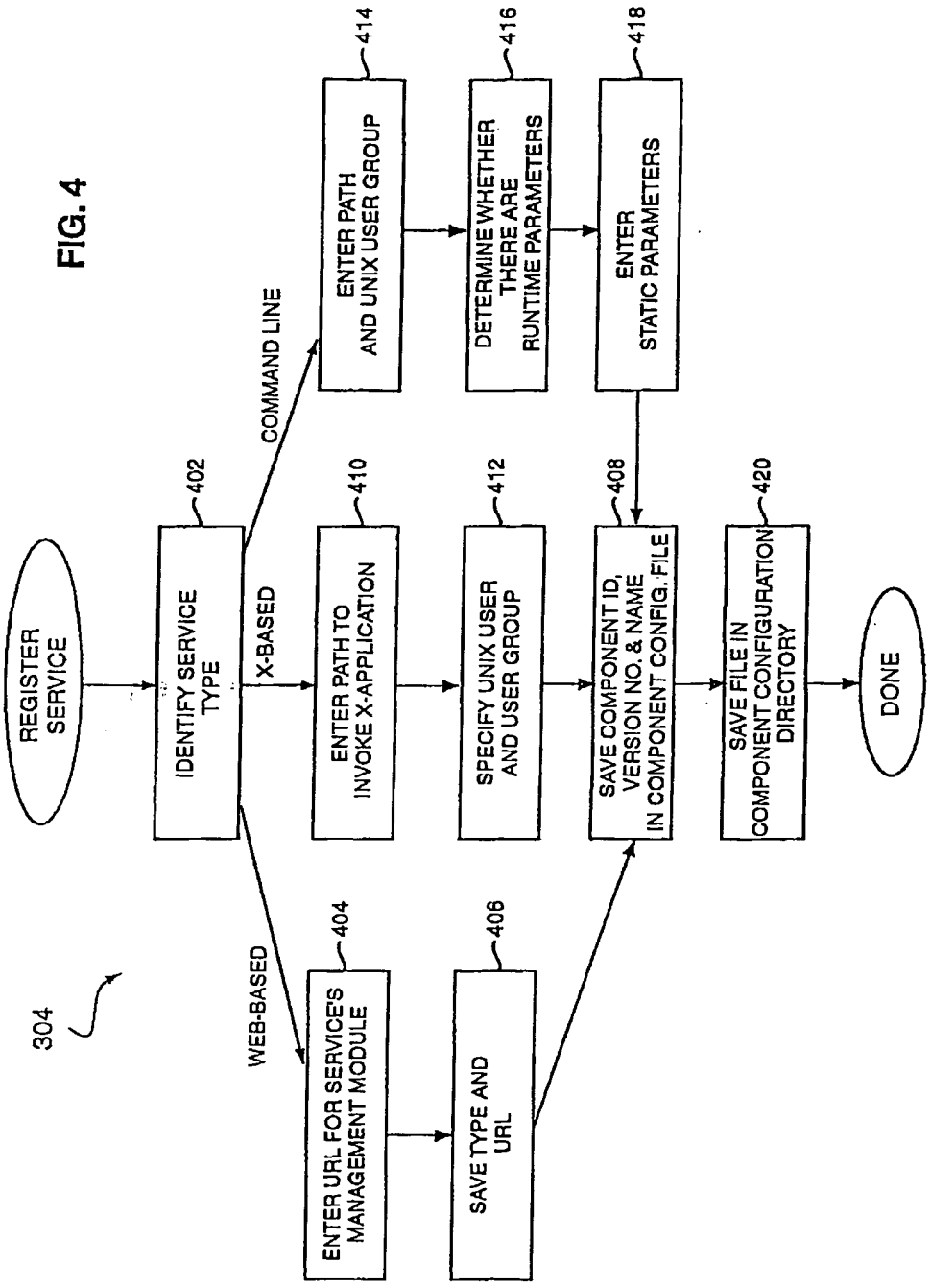

REGISTER NEW SERVICE
INSTALL SERVICE AND MANAGEMENT MODULE ON SERVICE HOST
302
REGISTER SERVICE ON SERVICE HOST (e.g. run mc_reg command)
304
USE MANAGMENT CONSOLE TO RUN "DISCOVER" ON SERVICE HOST
306
DONE

FIG. 3

FIG. 4


304
REGISTER SERVICE
IDENTIFY SERVICE TYPE
402
COMMAND LINE
X-BASED
WEB-BASED
ENTER PATH AND UNIX USER GROUP
414
DETERMINE WHETHER THERE ARE RUNTIME PARAMETERS
416
ENTER STATIC PARAMETERS
418
ENTER PATH TO INVOKE X-APPLICATION
410
SPECIFY UNIX USER AND USER GROUP
412
SAVE COMPONENT ID, VERSION NO. & NAME IN COMPONENT CONFIG. FILE
408
SAVE FILE IN COMPONENT CONFIGURATION DIRECTORY
420
DONE
ENTER URL FOR SERVICE'S MANAGEMENT MODULE
404
SAVE TYPE AND URL
406

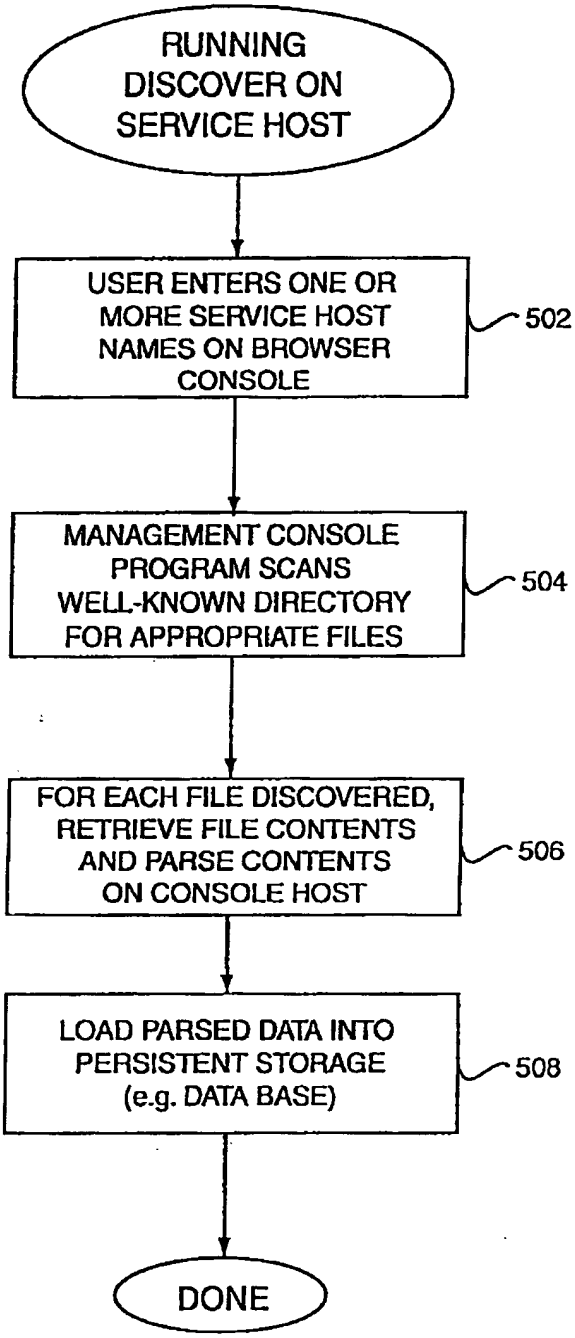

RUNNING DISCOVER ON SERVICE HOST
USER ENTERS ONE OR MORE SERVICE HOST NAMES ON BROWSER CONSOLE
502
MANAGEMENT CONSOLE PROGRAM SCANS WELL-KNOWN DIRECTORY FOR APPROPRIATE FILES
504
FOR EACH FILE DISCOVERED, RETRIEVE FILE CONTENTS AND PARSE CONTENTS ON CONSOLE HOST
506
LOAD PARSED DATA INTO PERSISTENT STORAGE (e.g. DATA BASE)
508
DONE

FIG. 5

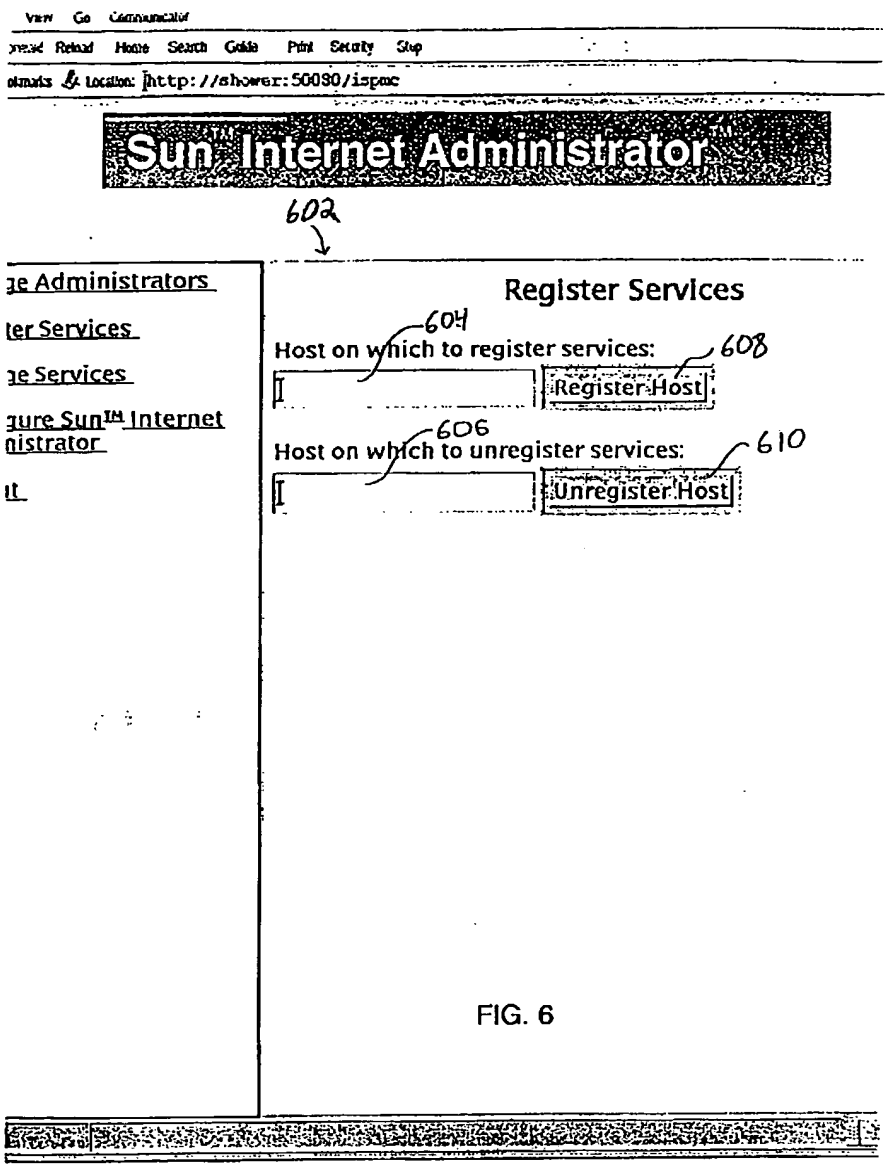

View Go Communicator
Reload Home Search Guide Print Security Stop
Bookmarks Location: http://shower:50080/ispmc
Sun Internet Administrator
602
ge Administrators
ter Services
ge Services
gure Sun™ Internet
nistrator
it
Register Services
604
Host on which to register services:
608
Register Host
606
Host on which to unregister services:
610
Unregister Host

FIG. 6

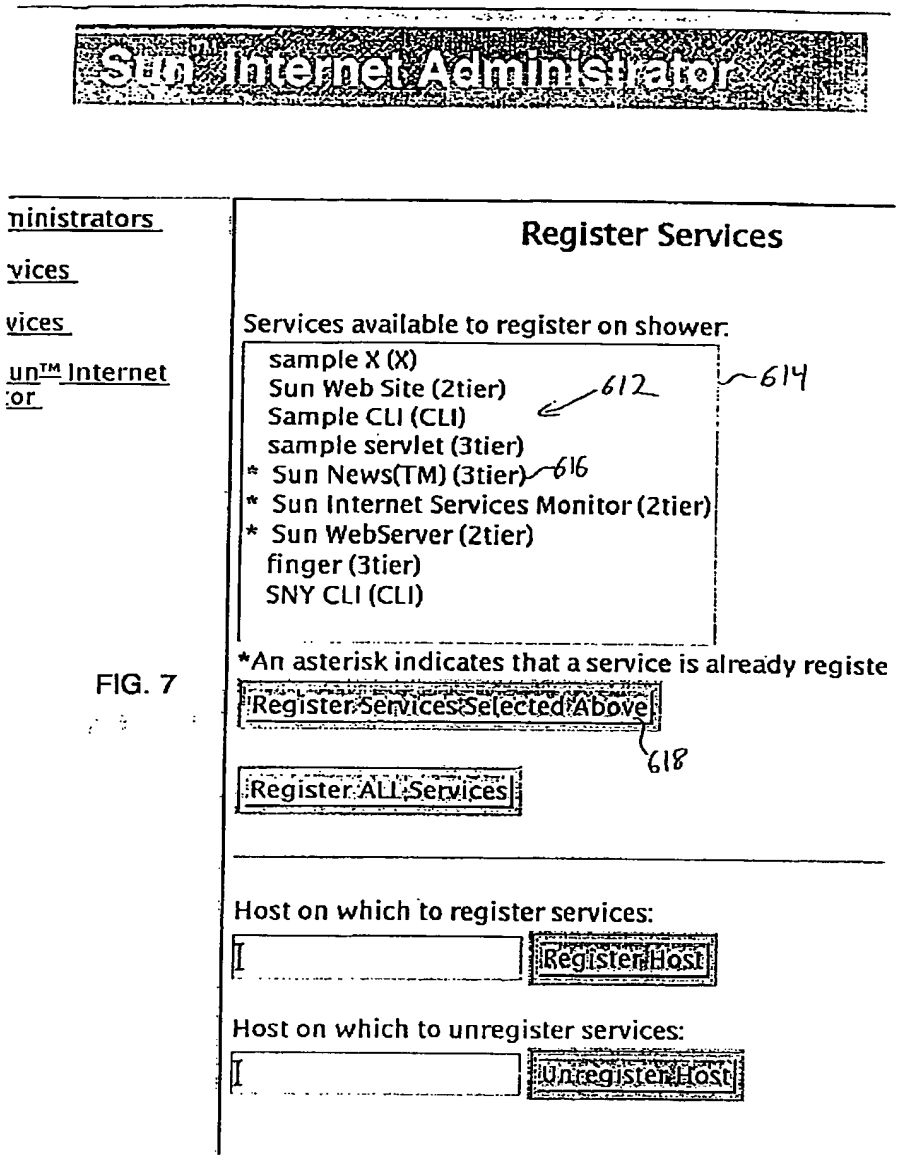

Sun Internet Administrator
ministrators
vices
vices
un™ Internet
or
Register Services
Services available to register on shower:
sample X (X)
Sun Web Site (2tier)
Sample CLI (CLI)
sample servlet (3tier)
* Sun News(TM) (3tier)
* Sun Internet Services Monitor (2tier)
* Sun WebServer (2tier)
finger (3tier)
SNY CLI (CLI)
612
614
616
*An asterisk indicates that a service is already registe
Register Services Selected Above
618
Register ALL Services
Host on which to register services:
Register Host
Host on which to unregister services:
Unregister Host

FIG. 7

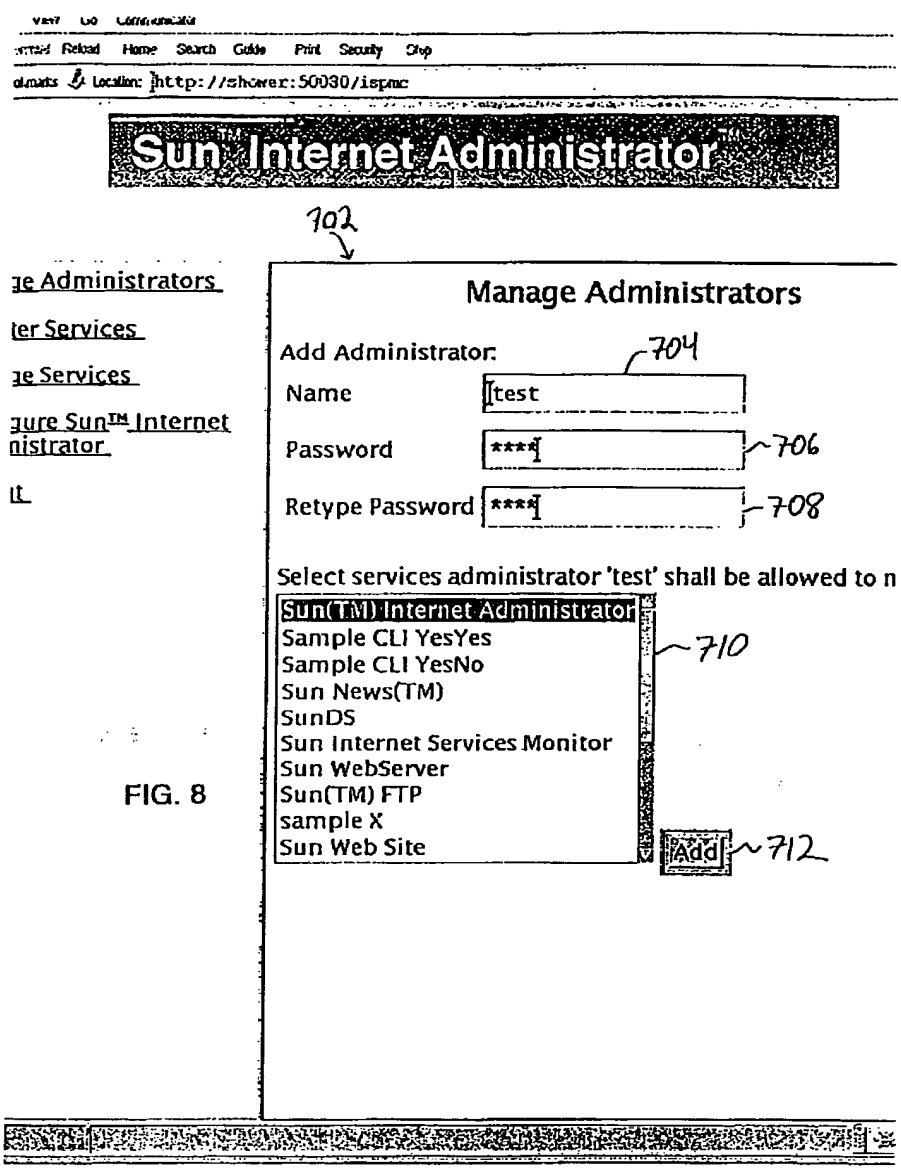

Reload Home Search Guide Print Security Stop
Location: http://shower:50030/ispmc
Sun™ Internet Administrator
702
ge Administrators
ter Services
ge Services
gure Sun™ Internet nistrator
it
Manage Administrators
Add Administrator:
704
Name
test
Password
706
Retype Password
708
Select services administrator 'test' shall be allowed to n
Sun(TM) Internet Administrator
Sample CLI YesYes
Sample CLI YesNo
Sun News(TM)
SunDS
Sun Internet Services Monitor
Sun WebServer
Sun(TM) FTP
sample X
Sun Web Site
710
Add
712

FIG. 8

FIG. 9

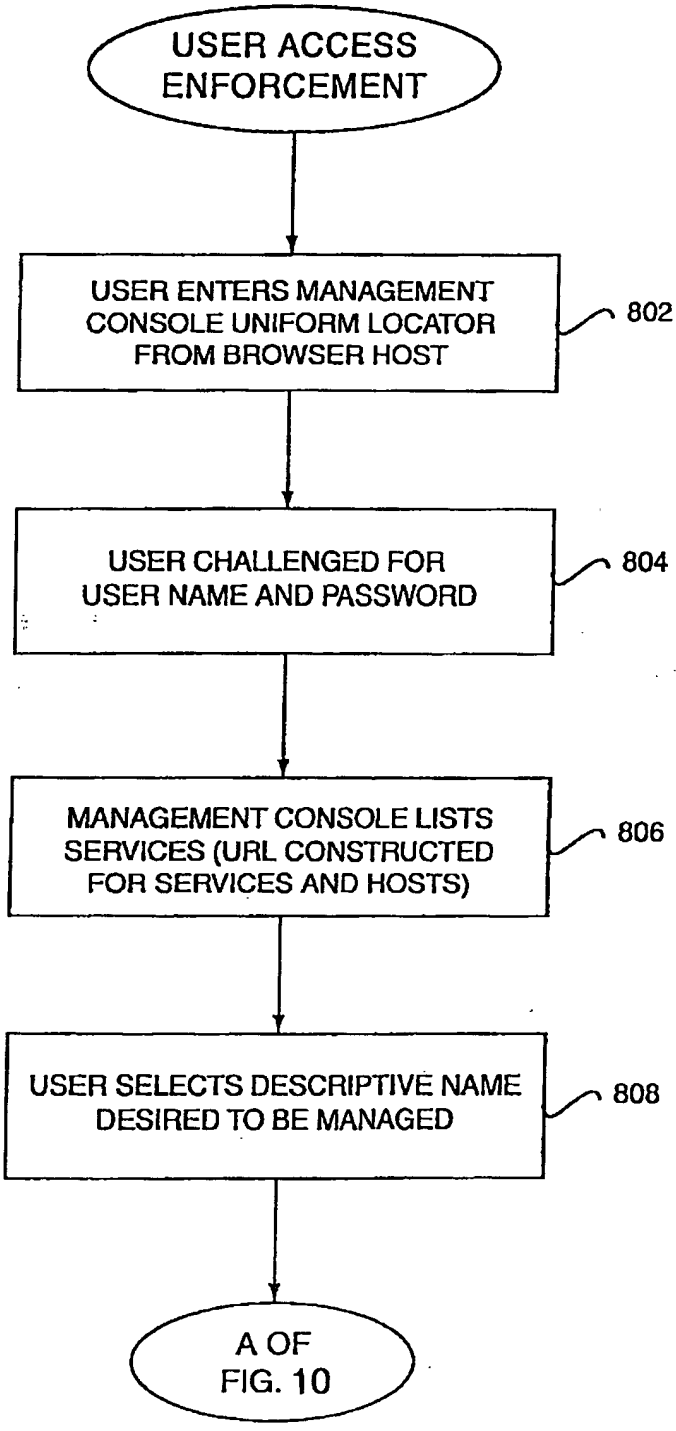

USER ACCESS ENFORCEMENT
USER ENTERS MANAGEMENT CONSOLE UNIFORM LOCATOR FROM BROWSER HOST
802
USER CHALLENGED FOR USER NAME AND PASSWORD
804
MANAGEMENT CONSOLE LISTS SERVICES (URL CONSTRUCTED FOR SERVICES AND HOSTS)
806
USER SELECTS DESCRIPTIVE NAME DESIRED TO BE MANAGED
808
A OF FIG. 10

FIG. 10

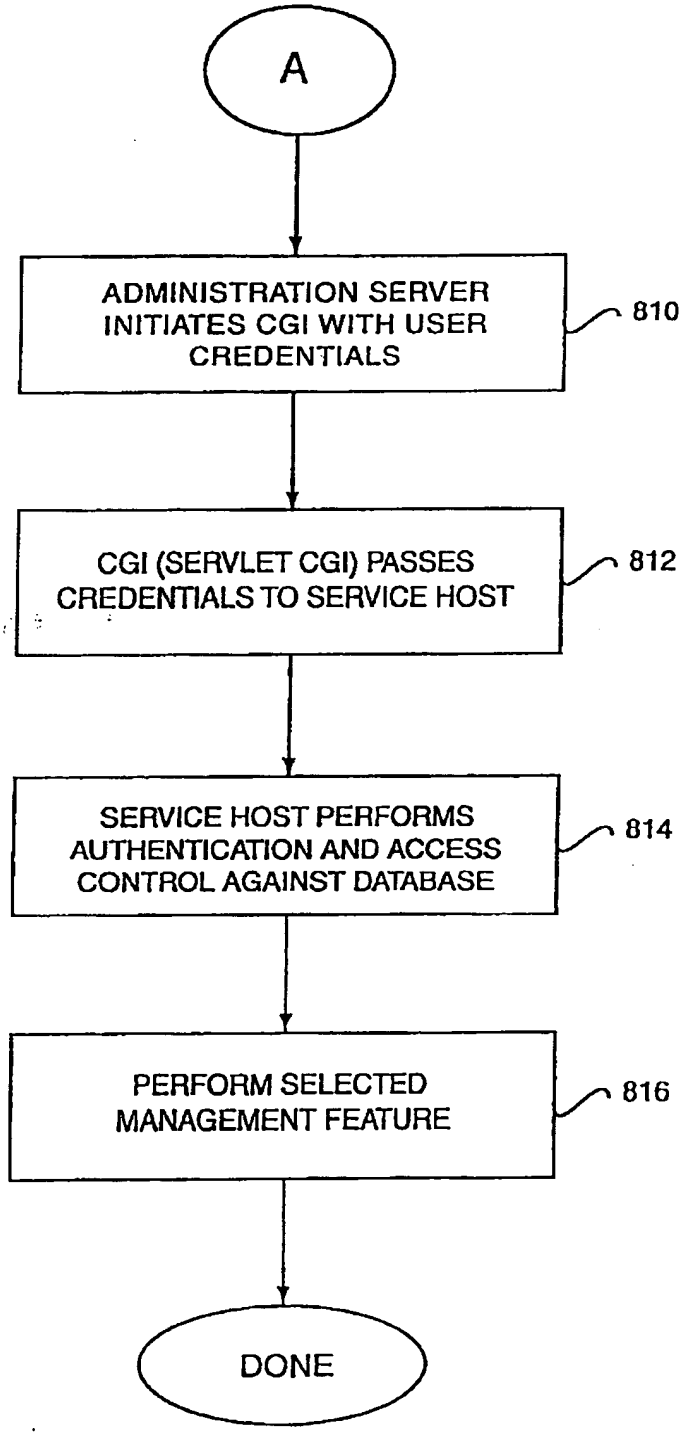
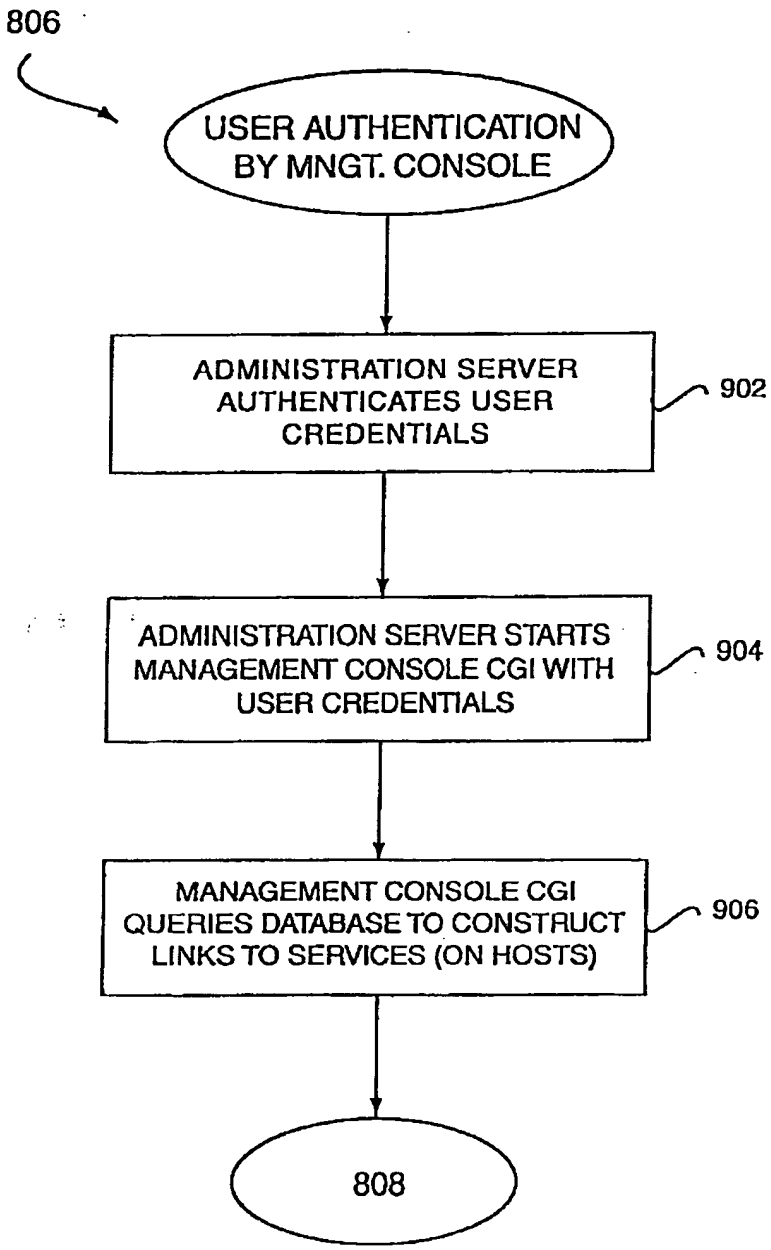

A
ADMINISTRATION SERVER INITIATES CGI WITH USER CREDENTIALS
810
CGI (SERVLET CGI) PASSES CREDENTIALS TO SERVICE HOST
812
SERVICE HOST PERFORMS AUTHENTICATION AND ACCESS CONTROL AGAINST DATABASE
814
PERFORM SELECTED MANAGEMENT FEATURE
816
DONE

FIG. 11


806
USER AUTHENTICATION BY MNGT. CONSOLE
ADMINISTRATION SERVER AUTHENTICATES USER CREDENTIALS
902
ADMINISTRATION SERVER STARTS MANAGEMENT CONSOLE CGI WITH USER CREDENTIALS
904
MANAGEMENT CONSOLE CGI QUERIES DATABASE TO CONSTRUCT LINKS TO SERVICES (ON HOSTS)
906
808

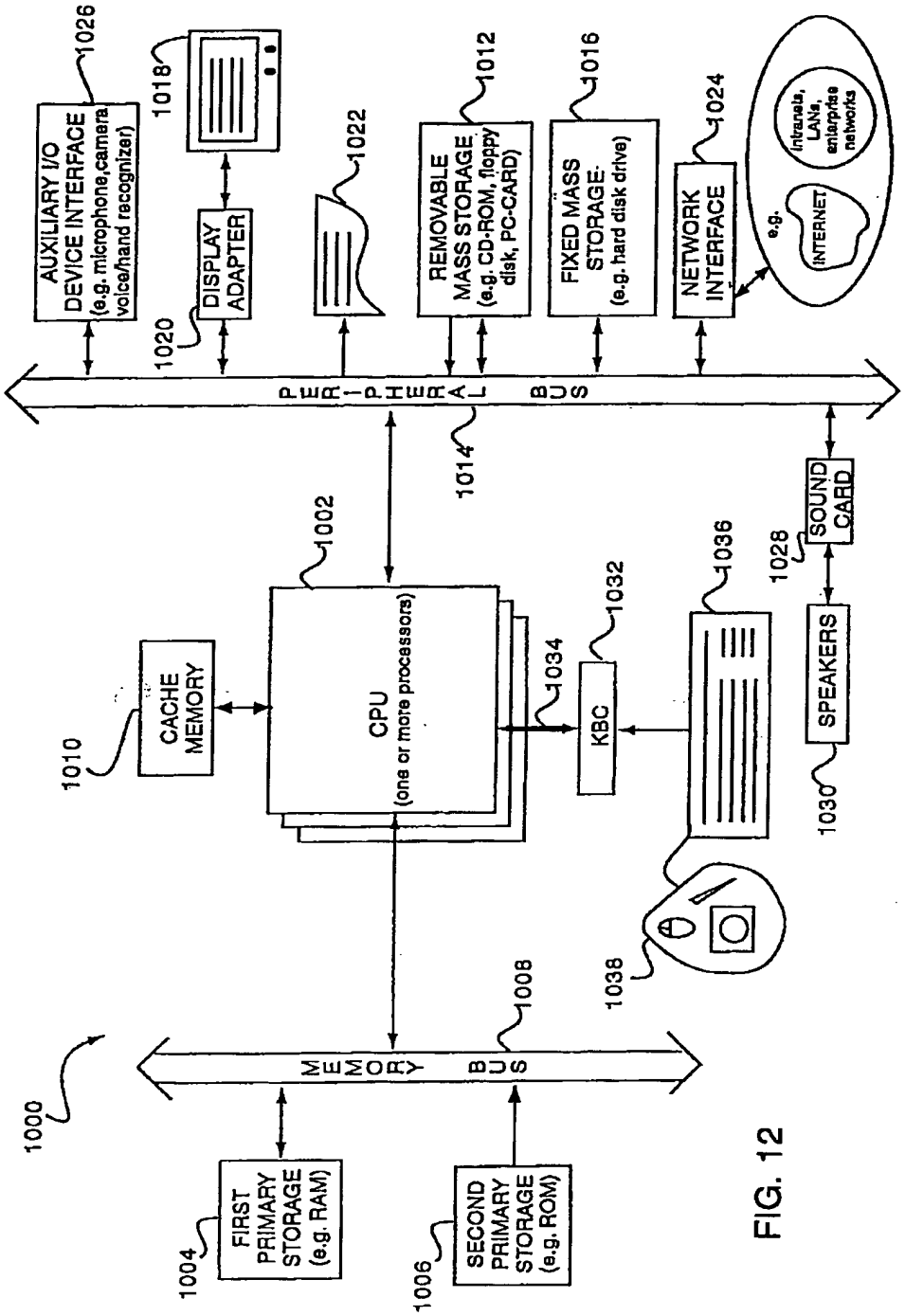

1000
MEMORY BUS
1008
FIRST PRIMARY STORAGE (e.g. RAM)
1004
SECOND PRIMARY STORAGE (e.g. ROM)
1006
CACHE MEMORY
1010
CPU (one or more processors)
1002
KBC
1034
1032
1036
1038
SPEAKERS
1030
SOUND CARD
1028
PERIPHERAL BUS
1014
AUXILIARY I/O DEVICE INTERFACE (e.g. microphone,camera voice/hand recognizer)
1026
DISPLAY ADAPTER
1020
1018
1022
REMOVABLE MASS STORAGE (e.g. CD-ROM, floppy disk, PC-CARD)
1012
FIXED MASS STORAGE (e.g. hard disk drive)
1016
NETWORK INTERFACE
1024
e.g.
INTERNET
intranets, LANs, enterprise networks

FIG. 12

1 ABSTRACT

A method and apparatus of securing access to a service manager for the administration of services residing on multiple service host computers from an administration server computer is described. A user identifier, such as a user name, and a corresponding password are provided to the service manager. The user identifier is associated with a system administrator having administrative access to the services. The service manager authenticates the user by comparing the user identifier and password against a list of user identifiers and corresponding passwords stored in a persistent memory. A list of services to which the system administrator has administrative access is derived from the data in persistent memory. When the system administrator makes a request to administer one or more services from the list of services, the administrator's access control is verified at the service host computers on which the requested services reside by examining access control data in the persistent memory.

Management files are transferred from the service host computers to the administration server computer thereby facilitating manipulation of the management files utilizing the service manager.

2 REPRESENTATIVE DRAWING

Fig. 2